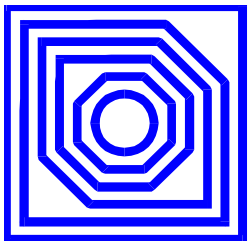
CHARON™-VAX version 3

for Windows® 2000 and Windows® XP host systems

User manual



User manual for CHARON-VAX/XM/XL version 3 for Windows 2000 and Windows XP. Includes CHARON-VAX/Industrial functionality with the exception of support for The Logical Company's BCI-2104, DCI-1100 and DCI-3100.

Document number: 30-16-010

Version: 4 February 2004

Email: vaxinfo@vaxemulator.com

Ownership Notice

Software Resources International SA owns all rights, including proprietary rights, copyrights, trademarks, and world-wide distribution rights to a methodology for the execution of VAX applications and system software by emulating VAX hardware components in software, henceforth referred to as CHARON-VAX.

The license for CHARON-VAX allows the execution of the software on a single host system or cluster. Possession, use, or copying of the software described in this publication is authorized only pursuant to a valid written license.

Software Resources International makes no representations that the use of the CHARON-VAX software as described in this publication will not infringe on existing or future patent rights, nor do the descriptions contained in this publication imply the granting of licenses to make, use, or sell equipment or software in accordance with the description.

Trademarks

The CHARON name with logo is a trademark of Software Resources International. VAX, Q-bus, VMS and OpenVMS are trademarks of Compaq Computer Corporation. Windows is a registered trademark in the United States and other countries, licensed exclusively through Microsoft Corporation, USA. Pentium is a registered trademark in the United States and other countries, licensed exclusively through Intel Corporation, USA. Athlon is a registered trademark of Advanced Micro Devices. All other trademarks and registered trademarks are the property of their respective holders.

Life support applications

The CHARON products of Software Resources International are not designed for use in systems where malfunction of a CHARON product can reasonably be expected to result in a personal injury. Software Resources International's customers using or selling our CHARON products for use in such applications do so at their own risk and agree to fully indemnify Software Resources International for any damages resulting from such improper use or sale.

Contents

Preface

The VAX processor family made by Digital Equipment Corporation has largely earned its place in the history of computing. Over two decades of development and sales exposed millions of computer users to these well-designed and reliable systems, which were even copied in the former East block as the best architecture available. But technology has moved on, and the quest for ever faster, smaller and cheaper components has made VAX hardware obsolete.

While system hardware will eventually wear out, fall apart, burn up or simply stand in the way, its software can remain to be of significant value to its users. However, application migration to another system architecture is not trivial and might even be impossible if no source code is available.

Modern computer systems can do complex tasks much faster than their ancestors of two decades ago. That makes it feasible to design a very precise software model of computer hardware in such a way that it can execute the original binary code but on a new host platform. CHARON-VAX/XM/XL provides such a model, providing models of various VAX and MicroVAX systems supporting Q-bus systems and SCSI systems with a and a range of Peripheral devices in a selection of VAX models supporting up to 512 MB VAX memory. The peripherals include serial lines, high performance SCSI disks, SCSI tapes, disk images, SCSI or ATAPI CD-ROM support, Ethernet adapters and 3.5 inch floppy support.

CHARON-VAX/XM/XL is designed to replace mid-range and large single CPU VAX systems and clusters with Windows servers. There is no need to modify, convert or upgrade your existing VAX/VMS operating system or VAX applications. With its high performance CPU option and the ability to form VMS clusters, it is an efficient

replacement product for departmental VAX servers that handle a heavy network and/or disk I/O load.

CHARON-VAX version 3

This manual has been updated for CHARON-VAX version 3.

Version 3 introduces a new range of high reliability small and mid range Q-bus systems that are available as part of the standard CHARON-VAX/XM/XL products.

CHARON-VAX/XM in addition to the MicroVAX 3100 model 96 it now includes MicroVAX 3600, MicroVAX 3900, VAX 3600, VAX 3900, and a VAX4000 model 106.

CHARON-VAX/XL in addition to the MicroVAX 3100 model 98 now includes everything available in CHARON-VAX/XM and a VAX4000 model 108.

Version 3 offers all of these new models as individual executables that may be run from the CHARON-VAX launcher or run as standard Windows Services.

Notes for CHARON-VAX/Industrial version 1 users

CHARON-VAX/XM (Plus) version 3 delivers a quality upgrade path for customers who run CHRON-VAX/Industrial (Plus). It includes the popular MicroVAX 3600 CPU and offers new versions of the MSCP, TMSCP and DHV11 functionality.

CHARON-VAX/XM (Plus) version 3 does NOT support Qbus adapters.

Organization of this manual

Chapter 1, *Overview*, introduces CHARON-VAX/XM/XL and describes its main features.

Chapter 2, *CHARON-VAX/XM/XL installation*, explains how to install and initially configure CHARON-VAX, how to install VAX/VMS on it, and how to transfer your existing VAX applications to CHARON-VAX/XM/XL.

Chapter 3, *CHARON-VAX/XM/XL configuration,* provides detailed information and examples of configuration parameters for CHARON-VAX/XM/XL.

Chapter 4, *Creating a simple configuration,* explains how to build a simple configuration file from scratch.

Chapter 5, *Utilities*, describes the CHARON utilities: Launcher, Service Installer, Service Manager and Configuration Wizard.

Appendix A contains examples of configuration files.

Appendix B describes the CHARON-VAX/XM/XL console

Appendix C refers to the use of Pathworks 32 (an HP product for VT525 and X-terminal emulation).

Appendix D contains some frequently asked questions and answers to assist with trouble-shooting.

Appendix E contains notes on the use of VAX/VMS.

Further information about CHARON-VAX/XM/XL:

See the CHARON-VAX Utilities and interconnects manual number: 30-16-015 for details of further CHARON-VAX utilities.

The web site www.charon-vax.com provides information about the CHARON-VAX products including a set of application notes.

When CHARON-VAX is running on Hewlett Packard products, the transfer license maintains the Compaq/HP VAX software warranties. Further information is available at:

http://www.openvms.compaq.com/openvms/sri-charon-vax-emulator.html

If you have questions or suggestions, you can contact us as follows:

Send an Email to **vaxinfo@vaxemulator.com,**

contact us by phone at **+41 22 794 1070**, or

send a FAX to **+41 22 794 1073**

Conventions

The following conventions are used in this manual:

Notation	*Description*
$ or >	The dollar sign or the right angle bracket in interactive examples indicates operating system prompt.
User Input	Bold monospace type in interactive examples indicates typed user input.
<path>	Bold monospace type enclosed by angle brackets indicates command parameters and parameter values.
Output	Monospace type in interactive examples indicates the output the command responds with.
[]	In syntax definitions, brackets indicate items that are optional. This convention is not applicable to the syntax of CHARON-VAX/XM/XL configuration files.
…	In syntax definitions, a horizontal ellipsis indicates that the preceding item can be repeated one or more times.

Chapter 1 Overview

1.1 Introduction

CHARON-VAX/XM and CHARON-VAX/XL are software packages that provide an exact mathematical model of VAX system hardware. They emulate all essential components (CPU, disks, Ethernet adapter, memory subsystem) required to execute a VAX operating system and applications in the same way as on a hardware VAX.

It is a bit confusing: CHARON-VAX/XM/XL is a **software** package, but it emulates VAX **hardware** (not a specific VAX operating system) on top of which you run your VAX operating system (usually VAX/VMS) and your VAX applications.

As the emulator executes the same binary VAX code as the original hardware, your existing VAX software will run without any modifications. There are no source code conversion costs, and the existing application behavior and user interface are preserved. When running CHARON-VAX/XM/XL, you have effectively created on its host system a virtual VAX system. You can use that system as a hardware VAX, run you copy of VAX/VMS, network or cluster it with other systems, all without having to maintain your VAX hardware.

An additional advantage of CHARON-VAX/XM/XL over a hardware-based emulator is the scalability with its host system. The CHARON-VAX/XM/XL performance is proportional to the CPU clock frequency of your host system, and every time you move to a faster host system your ‘virtual VAX’ will also get faster.

1.2 CHARON-VAX/XM/XL system components

CHARON-VAX/XM provides

An extended MicroVAX 3100 model 98 server configuration with a MicroVAX 3100 model 96 option. This includes the NVAX CPU, two SCSI controllers (each supporting up to 56 disk or tape drives, disk images, CD drives and Floppy drives), up to 128 MB of VAX memory, and it's integrated Ethernet controller (SGEC). SCSI disks can be much larger than the maximum size (4.3 GB) originally shipped with the MicroVAX 3100 model 96.

MicroVAX 3600 & 3900 are small Q-bus VAX supporting up to 64 MB of main memory. They are offered in two formats the MicroVAX 3600 or 3900 and the VAX 3600 or 3900. The VAX 3600 and 3900 are server systems. Both the MicroVAX 3600 and 3900 can also be configured as Workstations by loading the VCB02 stu.

The VAX4000 model 106 is a general purpose mid-size VAX supporting the same memory (128MB) and peripherals as the 3100 model 96 plus support for Q-bus peripherals. (DHV11/MSCP/TMSCP)

CHARON-VAX/XL is an extended memory system supporting all of the models of CHARON-VAX/XM with the addition of an extended memory of 512 MB. This extended memory provides support for the MicroVAX 310 model 98 and the VAX 4000 model 108.

Note that with the MicroVAX 3100 model 96/98 there are certain limitations imposed by VMS. Like the real VAX the CHARON-VAX MicroVAX 3100 model 96/98 needs a VMS version of 5.5-2H4 or later.

As a server configuration, CHARON-VAX/XM/XL does not emulate graphics hardware. Graphic and character cell terminals can be

connected via Ethernet. Pathworks32 is recommended as a flexible tool for X-terminal and VT525 emulation, in addition to its Windows VMS file access utility (see Appendix C).

1.3 Hardware compatibility

CHARON-VAX/XM/XL has been tested with AXE, the VAX architecture compatibility verification tool used by Digital in the design of VAX hardware. Compaq/HP has verified CHARON-VAX compatibility with AXE, and provides VMS and layered product licenses for the transfer from hardware VAX to the CHARON-VAX emulator. When CHARON-VAX is running on Compaq/HP products, the transfer license maintains the Compaq/HP VAX software warranties. Further information is available at:

http://www.openvms.compaq.com/openvms/sri-charon-vax-emulator.html

The VAX replacement components provided by CHARON-VAX/XM/XL are designed to operate like their hardware equivalents. CHARON-VAX does not include specific hardware diagnostic modes not used in normal operation. Time delays that would be required for diagnostic software to correctly verify mechanical device behavior are not included.

CHARON-VAX/XM/XL can in principle execute any VAX operating system or binary application within the scope of its components. However, it is optimized for its main application area: the replacement of larger single CPU and clustered VMS servers in administrative applications. CHARON-VAX/XM/XL has been tested to work with VMS 5.5-2H4, VMS 6.2 and VMS 7.3. See the notes on VMS in the appendices. CHARON-VAX/XM/XL is not designed to replace VAX systems in real-time process control applications.

When running OpenVMS/VAX, CHARON-VAX/XM/XL can replace most modern VAX systems, also outside the VAX 3100 models

described above, as VMS strictly separates the applications from the underlying hardware characteristics. CHARON-VAX/XM/XL systems running OpenVMS can be networked or clustered with other OpenVMS systems running on CHARON-VAX/XM/XL or on VAX or Alpha hardware.

1.4 Product versions

The difference between CHARON-VAX/XM and CHARON-VAX/XL is the emulated VAX memory size: CHARON-VAX/XM has a maximum of 128 MB; CHARON-VAX/XL emulates the full 512 MB of the NVAX CPU.

Both product variations are available in two CPU performance levels: The standard VAX CPU emulator or the Dynamic Instruction Translation (DIT) mode (see chapter 3.2).

The licenses listed below will not run on a host system with more than 2 CPU's. Time limited licenses; upgrades from CHARON-VAX/XM to CHARON-VAX/XL and from the standard CPU to the DIT CPU model are available.

CHARON-VAX/XM:

VAX CPU	Max. VAX memory	Product code
Standard	128 MB	CHVX-021-PD-WI
Plus (DIT)	128 MB	CHVX-221-PD-WI

CHARON-VAX/XL:

VAX CPU	Max. VAX memory	Product code
Standard	512 MB	CHVX-021-PF-WI
Plus (DIT)	512 MB	CHVX-221-PF-WI

1.5 Performance

The CHARON-VAX CPU performance scales with the clock frequency of the host CPUs provided the host memory latency is increased by the same factor. It also depends on the VAX CPU mode (standard or DIT). On a dual 2.4 GHz Xeon 4 system, the DIT VAX CPU emulation provides ± 20 MIPS; the integer performance is around 120K Dhrystones (± 20%). The standard VAX CPU (without DIT) emulator provides about one third of this.

Depending on the host platform, the CHARON-VAX/XM/XL DIT VAX CPU performance can match that of a VAX 7710 CPU. The disk I/O performance with modern SCSI disks exceeds that of any hardware VAX system.

1.6 Host system requirements

The CHARON-VAX/XM and CHARON-VAX/XL emulators are available for Windows 2000 and XP systems with dual Intel or AMD CPUs. Although the software will run on a single CPU host system, a dual CPU host system is required for adequate performance, especially if the DIT CPU optimization or intensive I/O is used. A single CPU with Hyper-Threading enabled will meet the needs of a second CPU in many cases. Note that you must **disable** Hyper-Threading on a dual CPU system with Hyper-Threading, as CHARON-VAX will refuse to start if it sees more than two logical CPU's (four in this case).

As an exception, a single CPU host system can be used for lightly loaded systems if VAX/VMS is used in conjunction with the special VMS "Idle loop detection Package". See the utilities chapter for the use of this package that establishes a link between the VMS scheduler and the emulator to optimize the use of a single CPU host.

To correctly function, CHARON-VAX/XM/XL requires a host system with a CPU clock frequency of at least 1 GHz. At startup, the emulator will issue a warning if it detects a CPU with a lower frequency, and will terminate execution when the detected frequency is lower than 500 MHz.

Note that the host system must be dedicated to CHARON-VAX/XM/XL for reliable performance. In a typical system the emulated VAX CPU will load one host CPU for 100%. The second host CPU handles the emulation of the remaining emulated VAX components, in particular disk and network I/O. The second host CPU retains enough capacity for the Windows system resources and a local (X) terminal emulator.

The table below lists the host system memory requirements for each VAX CPU emulation mode and the amount of VAX memory:

Product	**Product code**	**VAX memory emulated**	**Host memory required**
CHARON-VAX/XM	CHVX-021-PD-WI	128MB	256 MB
CHARON-VAX/XM Plus	CHVX-221-PD-WI	128MB	512 MB
CHARON-VAX/XL	CHVX-021-PF-WI	512MB	1 GB
CHARON-VAX/XL Plus	CHVX-221-PF-WI	512MB	1 GB

CHARON-VAX disks can be assigned to host system SCSI drives (not formatted by Windows) or are represented as a disk image file. Disk images are standard local or remote Windows files. The CHARON-VAX/XM/XL executable requires approximately 3 MB disk space and can be stored on the IDE drive.

CHARON-VAX/XM/XL can also use SCSI tapes, a SCSI or ATAPI CD-ROM or PC floppy drives. CHARON-VAX/XM/XL requires a

dedicated Ethernet adapter for networking and a free USB port for the USB license key.

Chapter 2 CHARON-VAX/XM/XL installation

This chapter explains how to install CHARON-VAX/XM/XL, how to configure a basic system in CHARON-VAX/XM/XL allowing installation a VAX operating system, and how to transfer your applications.

2.1 Before you start

Please read the Release Notes…

The CHARON-VAX/XM/XL distribution kit contains release notes, and we recommend that you read these notes before installing and using the product. The release notes may contain information about changes to the application since the publication of this manual.

Check the contents of the software distribution kit

The packing list specifies the number and contents of your media. Be sure to verify the contents of your kit with this information. If your kit is damaged or if you find that parts of it are missing, please contact Software Resources International or your distributor.

2.1.1 Installation overview

The CHARON-VAX/XM/XL installation kit is provided on CD. The Installation menu is started by running the setup.exe procedure in the root directory of the CHARON-VAX/XM/XL distribution CD, or will start automatically if you have “autorun” enabled on your system. The procedure will install the driver for the "Hardlock" license key. The key must be installed correctly before you can proceed with the installation of the CHARON-VAX/XM/XL executable and utilities. Note that if you are upgrading from an older version of CHARON-

VAX your license may not be recognized. Uninstall the Hardlock driver (Control panel -> Add/Remove programs) and run the CHARON-VAX installation again. You can remove both the CHARON-VAX/XM/XL Emulator and the Hardlock device driver using the standard Windows procedure:

`Start > Settings > Control Panel > Add/Remove programs.`

2.1.2 Installing the Hardlock license key.

CHARON-VAX/XM/XL for Windows is protected with a “Hardlock” USB license key. You must use a computer system with a free and properly installed USB port as without the license key the software cannot be installed or run. The Hardlock device driver must be installed as the first step of the installation process, before connecting the key to the PC.

Note: If you inserted the key before its driver is installed the Windows operating system will automatically look for the USB driver. In that case, do not cancel the USB installation but select the top directory of the CHARON-VAX/XM/XL installation CD, which contains the required AKSUSB.INF and the AKSUSB.SYS files. Once the USB driver is completed, continue with the license key installation item on the CHARON-VAX/XM/XL installation CD.

When CHARON-VAX/XM/XL starts, the license checking procedure takes a few seconds. When checking, a verification message is displayed. If you remove the license key while CHARON-VAX/XM/XL is running a warning message is given after a few minutes, and you have a maximum of 10 minutes to save your files and shut down your replacement VAX system. Reapplying the key during this period will not help, as the termination process cannot be undone.

Please take care with the license key, as it represents the full value of the license and will not be replaced if lost. The mean time between failure of the license key is approximately 100 years. If the license

key is broken a replacement will be supplied on return of the broken key. If you want to ensure continuous processing in case of critical applications schedules and where there is a risk of license key damage please contact your supplier or Software Resources International for more information.

2.1.3 Installing the CHARON-VAX/XM/XL executables

Unless directed elsewhere, the installation procedure will place the software in the directory C:\Program Files\CHARON Emulators.

2.1.4 Network installation

CHARON-VAX/XM/XL requires a dedicated Ethernet adapter. Since in principle your Windows host system should be dedicated to CHARON-VAX/XM/XL, you do not need to install an additional adapter, as you can use the same adapter for occasional tasks in Windows (e.g. network backup) when CHARON-VAX/XM/XL is not running.

CHARON-VAX/XM/XL network installation and configuration involves three steps. The first step installs the CHARON-VAX specific NDIS5 packet driver. The second step is to obtain the NIC device name you want to use for networking. The third step inserts the NIC device name in the CHARON-VAX/XM/XL configuration file.

1. Installing the NDIS5 packet driver.

Open the ‘Network Connections’ applet from the Start | Settings menu. Select the Local Area Connection to be used with CHARON-VAX/XM/XL, click the right mouse button and open the ‘Property’ applet. Switch to the ‘General’ pane and check the list of the installed protocols.

If you have previous versions of the Charon's packet protocol drivers, uninstall these first. To do so, select the protocol driver, click uninstall and follow the Windows questions. Don't forget to reboot the system when complete.

If you do not have a previous NDIS5 driver installed, click the 'Install' button, select 'Protocol' from the list of the 'Network Components' and click the 'Add…' button. In the 'Select Network Protocol' applet, click the 'Have Disk' button. In the 'Install from Disk' applet select the directory in the CHARON Emulators folder under drivers with name ndis5. The Applet should automatically pick up the packet.inf file with the driver information. Click 'Ok' to install the driver. Don't forget to reboot the system after driver installation.

After system reboot, come back to the 'Network and Dialup connections' applet and check that 'Charon's Packet Protocol Driver (NDIS5)' is present and enabled in the list of the installed components. Disable all other components since CHARON must have exclusive access to this network card.

2. Obtain the NIC device name.

Open the Start | Settings | Network and Dial Up Connections configuration dialog. Select the connection you want to use for CHARON networking, click the right mouse button and open Connection's Properties applet. Check that the 'Charon Packet protocol (NDIS5)' connection is only one enabled. Remember or copy to the clipboard the device name presented in the 'Connect using:' edit box. You will need to write exactly this device name (case sensitive) in the interface option of the ndis5_packet_port instance.

3. Enter the ID in the configuration file.

Configure the ndis5_packet_port component in your configuration file by pasting the clipboard after the "=" sign in this load command:

```
load ndis5_packet_port eza_0 interface="…"
```

The next statement connects the logical port (named eza_0) to the EZA Ethernet controller of the replacement VAX 3100/9x:

```
set eza interface=eza_0
```

That is all that is normally required. Please see the section on configuring the Ethernet Adapter for a description of additional tuning parameters.

2.1.5 Configuring the replacement MicroVAX 3100 model 9x

After installation you must configure your CHARON-VAX/XM/XL system to install VAX/VMS. Assuming a default installation you can edit configuration file (charon.cfg is the default) with Notepad or any other text editor. This file is available in the CHARON-VAX target directory in the VAX3100\bin folder. The configuration file should be saved in text format. You can write your own configuration file based on the default.

Read the Chapter on Configuration to prepare a configuration file for the desired VAX system (within the limits of the CHARON-VAX hardware). The commented configuration file examples in Appendix A illustrate the structure and content of the configuration file parameters.

2.1.6 Starting CHARON-VAX

Start CHARON using one of the following methods.

Use the desktop shortcut to run CHARON-VAX with the default configuration file.

You may wish to create your own configuration file. In this case you can either create a new shortcut for your alternative configuration or

start CHARON-VAX/XM/XL from the console, using the following syntax:

>charon c:\my_cfg\my_configuration.cfg

Please go to VAX3100\bin to issue such a command or define a PATH variable so that the system can find the CHARON-VAX/XM/XL executable from any directory.

The CHARON-VAX Launcher allows you to select a configuration file to run and to view the errors or log file from the Launcher Window. Access the CHARON-VAX Launcher in the CHARON Start Menu. The CHARON-VAX Launcher is useful for debugging configuration changes. See the Utilities Chapter for a full description of the CHARON-VAX Launcher.

2.1.7 Running CHARON-VAX as a service

Once the desired configuration has been tested install the configuration as a service using the CHARON-VAX Service Installer and manage the initiation of that service using the CHARON Service Manager. See the utilities Chapter for details of these utilities. The separation of Service Installer and Service manager allows the system manager to manage the content of available services and allows operational staff to run the services without risk of making unauthorized or unintentional changes to the installed services. To make it impossible to modify installed services remove the Service Installer program and Service Installer DLL.

2.2 Transferring data to CHARON-VAX

There are several ways to transfer data from a MicroVAX to CHARON-VAX/XM/XL:

2.2.1 Using the Local Area Network

Load your Operating System from the manufacturer's original media. Configure the Network to add your CHARON-VAX to your Network with a unique address and use DECnet or TCP/IP to copy your applications and data to your CHARON-VAX. If for any reason building an Operating system from scratch is a problem call your CHARON-VAX sales contact for help in preparing a temporary alternative. Once you have a CHARON-VAX connected to your Network you may use standard utilities to copy across the required data. Before copying the data you will have to configure CHARON-VAX with adequate free space on disks or on disk images that can be created with the utility MKdisk.

2.2.2 Using a physical disk drive

You can remove a SCSI disk from your VAX system and reconnect it to a SCSI adapter in your Windows system. Assign the SCSI disk in the CHARON-VAX/XM/XL configuration file to a disk controller and it becomes a disk drive in the CHARON-VAX. If the SCSI disk is a bootable VAX/VMS disk you can boot the replacement MicroVAX 3100 model 96/98 from it.

Note: As the CHARON-VAX/XM/XL emulator is based on a MicroVAX 3100 model 96/98 there are certain limitations imposed by VMS. Like the hardware VAX system, the emulated MicroVAX 3100 model 98 needs VMS version 5.5-2H4 or later.

2.2.3 Using a tape

CHARON-VAX/XM/XL supports the connection of a SCSI tape drive to a SCSI adapter in your Windows system. Assign the tape drive in the CHARON-VAX/XM/XL configuration file to operate the tape drive as usual by the VAX operating system. This way you can boot from standalone tape to restore your system backup.

Older tape drives that do not have a Windows driver can still work with the CHARON-VAX. See chapter 3.6 and 4.3 for details. Note that the reliability of a physical tape connection depends on factors like SCSI controller type, tape drive model and host CPU speed. Tape connections are not guaranteed to work in all cases.

Chapter 3 CHARON-VAX/XM/XL configuration

At startup, CHARON-VAX/XM/XL assembles a complete 'virtual' VAX system by loading the CHARON-VAX CPU, memory, system bus and peripheral components, based on the information coded in the licensing mechanism.

'Assembling' and starting the system is automatic, but information is required about the location of the VAX disks, the Ethernet adapter and other configuration options. This information is listed in a configuration file, a text file read by CHARON-VAX/XM/XL at startup.

With this information, CHARON-VAX/XM/XL starts the execution of the internal diagnostics and subsequently, if the diagnostics complete correctly, starts the boot sequence of the indicated VAX boot disk.

To run CHARON-VAX/XM/XL use "Start->Programs->CHARON Emulators->CHARON Launcher" (see the chapter 5.1). It is also possible to run CHARON-VAX/XM/XL from a DOS console window as follows:

>charon charon.cfg

Use the CHARON Launcher to diagnose and correct configuration errors.

Once your CHARON-VAX/XM/XL configuration is stable it may be installed as a Windows system service and the service may be started and stopped manually or automatically using the CHARON Service Manager.

When CHARON-VAX/XM/XL starts it creates an icon on the taskbar menu. Clicking on this icon with the right button of your mouse gives 2 options: "Stop Emulator & Exit" and "Halt (Reset)" which is the analogue of the HALT button on a physical VAX. "Stop Emulator & Exit" is disabled if CHARON-VAX/XM/XL is running as a service.

See the Utilities Chapter for further details on configuring, installing running and managing CHARON-VAX/XM/XL.

3.1 Configuration file parameters

In a CHARON-VAX/XM/XL configuration file the device settings are indicated with the SET command. Each configuration parameter has a unique name and gets a value assigned to it. For example:

```
set RAM size = 128
```

Sets the amount of memory to 128 MB (if this is within the limits allowed by the product license). Parameters used in a SET command can be of the following types:

- **Boolean**. Valid values are **true**, **false**. A value of true means that the option is enabled, a value false means that the option is disabled.
- **Identifier**. Used to set devices instances and their parameters.
- **Character**. Values of this type are character strings enclosed by quotation marks ("string parameter").
- **Numeric**. Where numeric data is required, it can be entered in the following formats:
  - Octal: as a number starting with 0; use the symbols 0 – 7.
  - Decimal: a number starting with 1 - 9.
  - Hexadecimal: a number starting with 0x; 0 - 9 and a – f. Example: 0x1234abc

3.2 Session

Use the session parameter to define the CHARON-VAX/XM/XL model to be run and how to save the log file. Options:

Set parameter	Type	Value
hw_model	Character	Defines the CHARON-VAX model to be run. Permitted values: CHARON-VAX/XL loads a "MicroVAX_3100_Model_98" CHARON-VAX/XM loads a "MicroVAX_3100_Model_96"
Log	Character	A string specifying the file name for the CHARON-VAX log.
log_method	Character	A string specifying whether the previous session log, if present, is to be overwritten or appended. Possible values are "APPEND" or "OVERWRITE". Default value is "OVERWRITE". If the value specified is other then "append" or "overwrite" then "overwrite" is used.

Example:
```
set session hw_model="MicroVAX_3100_Model_98"
```

Note: Specify *hw_model* before all the other settings in configuration file. hw_model cannot be omitted.

The CHARON-VAX log file can be stored as follows:

`set session <parameter>=<Value>`, for example:

```
set session log = “charon.log”
```

or

```
set session log="charon.log" log_method="append"
```

or

```
set session log="emulator.log" log_method="overwrite"
```

3.3 The emulated VAX CPU

For CHARON-VAX/XM/XL two VAX CPU models are available, the standard VAX instruction decoder and the optional high performance "Dynamic Instruction Translation" (DIT) model. The DIT option optimizes the VAX instruction interpretation. This significantly improves performance, but requires approximately four times as much host memory to store the optimized code.

As this optimization is performed dynamically during execution, it does not need to write optimized code back to disk, and provides its full capability instantly. The optimization does not compromise the VAX instruction decoding; CHARON-VAX remains fully VAX hardware compatible and completely transparent to VAX operating systems and applications.

Both CPU models pass the Compaq VAX Architecture (AXE) tests, which is the qualification for VAX instruction execution correctness.

The default VAX CPU mode is determined by the specific CHARON-VAX/XM/XL product license.

Set parameter	Type	Values
dit_enable	Boolean	**true** or **false.**

If the license permits DIT operation, **true** is default, otherwise **false** is default. For test purposes the DIT mechanism can be disabled with:

```
set cpu dit_enable=false
```

Note that

```
set cpu dit_enable=true
```

is ignored when the license does not permit DIT operation.

3.4 Setting memory size

The maximum amount of memory depends on the CHARON-VAX License. The /XM version is equivalent to a VAX 3100 model 96 that is limited to 128 MB; the /XL version provides the full 512 MB range of the VAX 3100 model 98. The logical for memory size is implicitly loaded as **`ram:`**

Set parameter	Type	Value
Size	Numeric	A value of N * 16 MB with N between 1 and the limits of the license.

For example:

```
set ram size = 128
```

3.5 The VAX console ports

CHARON-VAX/XM/XL implements the four standard VAX 3100/96 and 3100/98 serial console ports. The four-line serial line controller is identified within CHARON-VAX with the name **QUART**. The last line of the QUART (line[3]) is the VAX Console (known in VAX/VMS as OPA0).

The CHARON-VAX console ports can be configured to connect to an external terminal via a host system COM port or can be connected to a process via TCP/IP.

3.5.1 QUART Parameters

Set parameter	Type	Value

RTS[<line>]	Character	“On” - assert the RTS signal “Off” – clear the RTS signal (default) “DTR” - assert the RTS signal as soon as the DTR signal is asserted
DSR[<line>]	Character	“On” - always reports the DSR signal asserted “Off” - always reports the DSR signal deasserted “DSR” - use the DSR signal of the physical serial line (if configured) “CD”, “DCD”, “RLSD” - use the CD (carrier detect) signal of the physical serial line (if configured)
Communication[<line>]	Character	“ASCII” – for connection to terminals (default) “BINARY” - for serial lines carrying binary (packet) protocols, which are used mainly for communicating with PLCs

All the values in this table are case insensitive.

Example:
set QUART rts[1]=”DTR”
set QUART dsr[1]=”On”
set QUART communication[1]=”binary”

Note that the QUART does not allow applications to manipulate RTS directly (as there are no controlling bits in the registers).

Note that the line 2 of the QUART is the only one which might be used for connecting modems, therefore the DSR parameter for that

line (that is “dsr[2]”) is internally set to the appropriate value (“CD”) but can be changed from configuration file.

Both the RTS and DSR parameters were introduced mainly for supporting modems. Following the original design, the CHARON-VAX QUART provides input modem signals only for line 2. Therefore values of DSR parameter for lines other than 2 are not visible for application.

Connect via a COM port

Use the command **load physical_serial_line** to prepare for a connection to a COM port on the host system. The following **set** options are available:

Set parameter	Type	Value
Line	Character	Connects a CHARON-VAX serial line to a host system COM port. Example: “COM2:”
Break_on	Character	Defines a key (combination of keys) for break operation. “Ctrl-P” or/and “Break” – specify one of them or both separating with comma (“,”) “none” – No break key is defined The default value is “F5,Break” for line 3 of the QUART and “none” – for other lines.

Example:

```
load physical_serial_line DEF break_on="Ctrl-P,Break"
set DEF line[2]="COM2:"
```

or (short form):

```
load physical_serial_line DEF line[2]="COM2:"
```

Set **break_on** to “Break” to trigger the HALT condition in CHARON-VAX upon detection of SPACE condition on the associated COM port. Provided that the physical serial line connects a terminal to CHARON-VAX press the “Break” button on the terminal’s keyboard to generate SPACE condition on the serial line.

Set **break_on** to “Ctrl-P” to trigger HALT condition in the emulated VAX upon reception of Ctrl-P character (ASCII code 10 (hex)).

Note that only the console serial line (that is line[3] of the QUART) can trigger HALT conditions.

Connect via TCP/IP

Use the command **load virtual_serial_line** to prepare for a connection to 3rd party terminal emulators and other connections over TCP/IP. There are the following **set** options**:**

Set parameter	Type	Value
Host	String	The TCP/IP host name and, optional remote port to connect to.
Port	Numeric	Specifies the local TCP/IP port at which the virtual serial line waits for incoming connection requests.
Application	String	The application to run. For example a 3rd party terminal emulator connecting to a specified **port**.

break_on	String	Defines a key (combination of keys) for break operation. "Ctrl-P" or/and "Break" or/and "F5"– specify one of them or all separating with comma (",") "none" – No break key is defined The default value is "Break" for console line of QUART (line[3]) and "none" – for other lines.
Stop_on	String	Defines a key (combination of keys) to tell the emulator which external events could trigger STOP condition "F6" or/and "Application"– specify one of them or all separating with comma (",") "none" – No stop key is defined The default value is "F6" for console line of QUART (line[3]) and "none" – for other lines. Set to "Application" to trigger STOP condition when the associated application terminates. Use this option only for virtual_serial_lines configured for automatic application invocation (whose APPLICATION parameter specifies a valid application). Press the "F6" button to trigger STOP condition upon reception of the sequence "<ESC>[17~". Terminal emulators may send these sequences when pressing F6 button.

Example:

load virtual_serial_line GHI break_on=”Ctrl-P,F5”
set GHI port=10000 stop_on=”F6,Application”
or (short form):
load virtual_serial_line GHI port=10000

If you load a virtual_serial_line device note the following:

1. To use a 3rd part terminal emulator or similar program use the combination of parameters port and application as follows:

   load virtual_serial_line TTA0 port=10000 application=”tta0.ht”

   In this example CHARON-VAX/XM/XL creates port 10000 and waits for a connection. Then it starts HyperTerm (using the configuration file “tta0.ht”), which will connect to the port 10000 using the appropriate parameters in the file “tta0.ht”

2. To connect CHARON-VAX/XM/XL to a port on a specific host use the parameters host and port:

   load virtual_serial_line TTA0 host=”192.168.1.1” port=10000

   In this example CHARON-VAX/XM/XL connects to port 10000 of the host with TCP/IP address 192.168.1.1

   It is also possible to specify port on remote host (note that Charon always acts as a server). The syntax is:

   load virtual_serial_line TTA0 host=”192.168.1.1:20000” port=10000

   In this example Charon will accept connection on local port 10000 and connect to remote port 20000 of the host 192.168.1.1

Note that in the above examples ABC, DEF and GHI are logical names for the serial line emulation. These names are only used as a reference within a configuration file. They have no influence on the naming of the devices inside a VAX operating system. The names used can be helpful identifiers, use any character string you wish.

3.5.2 Link the 'logical' serial line with a QUART port.

Associate each loaded serial port with a line of the quart controller (up to 4) in the following way:

```
set quart line[<number of line>]=<instance name>
```

Example:

```
set quart line[0]=TTA0
```

If used in combination with the above load command for TTA0 this set command connects the first serial line (line[0]) of the QUART serial line controller to an terminal emulator connected via TCP/IP.

3.6 CHARON-VAX/XM/XL SCSI devices

CHARON-VAX/XM/XL provides two SCSI controllers. Hardware disks, Disk images, Hardware tapes, floppy devices and CD devices may be connected to these SCSI controllers. Each device has to be configured to connect to a specific SCI address in the CHARON-VAX.

Use the following emulated device types to map real peripherals to the emulated SCSI devices:

- ***virtual_scsi_disk*** – for disk image containers
- ***physical_scsi_device*** – for physical SCSI devices on the host. This instance type can be used also for any SCSI device, for example disk or tape drives.

- ***atapi_scsi_device*** – for any host ATAPI CD-ROMs.

- ***floppy_scsi_device*** - for host floppy disk drives.

CHARON-VAX/XM/XL uses SCSI disks connected to the Windows host system as VAX disks or disk containers that are represented in operating system environment as files. CHARON-VAX may join a VMS cluster. If planning a shared disk cluster please note that specific hardware is required and hardware iSCSI disks must be used, not disk images. Please contact us for more information. In line with the VAX 3100-96/3100-98 architecture, two SCSI controllers are supported (PKA and PKB) in CHARON-VAX/XM/XL, with 7 addresses each. The utility SCSI_CHECK (see chapter 5.3) helps to locate a SCSI device in the Windows host system.

Beyond the capabilities of the hardware VAX 3100/9x, CHARON-VAX/XM/XL has implemented extended SCSI addressing. Each of the seven device addresses of a SCSI controller can support up to eight disks. Thus the total number of disks supported becomes 2 Controllers*7 addresses*8 Disks, which gives a theoretical total of 112 disks.

SCSI devices with the same ID but different LUNs (logical units) appear in the VAX console with different names. The naming convention is as follows. Each SCSI device has the name of the form "xKct0n:", where 'x' stands for device type (D means disks, M means magnetic tapes, G is reserved by VMS for special purposes), 'c' stands for controller letter (A - the first controller, B - the second controller, ...), 't' stands for SCSI device ID (usually 0 through 6, and 7 is allocated by the controller itself), and finally 'n' stands for particular logical unit number LUN.

Most of the 'normal' SCSI devices have only one logical unit - 0. Therefore under normal conditions disks in VMS appear as DKA0 (which is really DKA000), DKA100, DKA200, ... As soon as there is a

disk device with LUNs 0 and 1, VMS handles them as, let us say, DKA300 and DKA301 respectively.

The boot ROM of the CHARON-VAX detects SCSI devices with multiple LUNs, and builds proper device names for them ('show dev' at the VAX console prompt to see a list). This list is passed to VMS when you start booting. Without help VMS only creates devices for logical units 0 for each device received from the boot ROM. One way to overcome this problem is to use the SYSGEN command as follows:

```
$ MCR SYSGEN CONNECT DKxxx/NOADAPTER
```

Where DKxxx stands for correct VMS name of the logical unit to be connected. You can find this name in the SRM console with “show SCSI” command. Add this command to the VMS SYSTARTUP_VMS.COM file to ensure that it is executed on each startup.

Also note that the following rules are applied for logical units.

1. Each SCSI device must implement logical unit 0.

2. A SCSI device must implement all logical unit numbers between the highest and the lowest numbers implemented.

Empty disk images can be created with the utility MKDISK. Disk images created with CHARON-VAX/Industrial can be read and written by CHARON-VAX/XM/XL. CHARON-VAX/XM/XL is able to boot VMS disk images of any VMS version of 5.5-2H4 or later. Disk images may be compressed using standard Windows data compression. The performance impact of data compression is minimal in most cases.

Physical SCSI disks used by CHARON-VAX/XM/XL must not contain a file system known to the Windows operating system; otherwise the

drive will not be available for use in CHARON-VAX/XM/XL. With the utility “Disk Management” (Control Panel | Administrative Tools | Computer Management | Disk Management) you can verify that Windows 2000/XP has not allocated the disks. If they do, use the disk management tool to delete the partition (i.e. remove the Windows file system) and thus release the disk.

Firstly load a SCSI device with the **load** command. At the same time specify the name of the device instance, the emulated SCSI bus to connect the device to and the SCSI identifier of CHARON device.

Load parameter	Type	Value
scsi_bus	Identifier	Name of emulated SCSI disk controller: **pka** or **pkb**
scsi_id	Numeric	A value between 0 and 7. ID number of the emulated SCSI device.

Note that there is no direct correspondence between the host hardware SCSI ID and these CHARON SCSI addresses. The correspondence between physical SCSI addresses on the host system and the CHARON SCSI bus ID has to be set in the configuration file;

Devices are loaded as follows:

```
load <instance type> <instance name> scsi_bus=<bus name> scsi_id=<number>
```

3.6.1 virtual_scsi_disk

Use **virtual_scsi_disk** for disk containers. This is the most convenient way of connecting disks to CHARON-VAX/XM/XL. It is possible to install an operating system on an empty disk container, to

restore a backup to a container or to create a target disk container with the MKIMAGE utility.

virtual_scsi_disk has the following parameter**:**

Set parameters	Type	Value
container[LUN]	Character	A string containing the full path to a disk container. LUN stands for logical unit number. It must begin with 0 and must have no gaps. If only a name of the disk container is specified CHARON-VAX/XM/XL will look for the container in the installation directory, in the directory VAX3100\bin

Example:

```
load virtual_scsi_disk pka_0 scsi_bus=pka scsi_id=0
set pka_0 container[0]=”C:\Charon\disk1.vdisk”
set pka_0 container[1]=”C:\Charon\disk2.vdisk”
```

If only one LUN is configured the LUN number can be omitted:

```
set pka_0 container=”C:\Charon\disk1.vdisk”
```

Note: The “geometry” parameter documented in earlier manuals is not required since the version 1.2 build 0 of CHARON-VAX/XM/XL.

3.6.2 physical_scsi_device

Use **physical_scsi_device** to connect any host SCSI device to the emulator.

physical_scsi_device has the following parameter**:**

Set parameters	Type	Value

container[LUN]	Character	A string containing the device name to map to the emulator. LUN stands for logical unit number. It must begin from 0 and have no gaps. If there is only one LUN it can be omitted. Default is 0.

Find the number of a SCSI disk device from the Device Manager. The syntax for SCSI disks is following:

```
load physical_scsi_device pka_0 scsi_bus=pka
scsi_id=0
set pka_0 container="\\.\PhysicalDrive4"
```

This example will associate the 4th unallocated SCSI drive in Windows 2000 or Windows XP with device **pka_0**.

CHARON-VAX/XM/XL supports SCSI tapes. All SCSI tape devices appear in Windows 2000 operating system as '\\.\Tape0', '\\.\Tape1', etc. Use these names in the configuration file to attach the SCSI controller to the tape devices of the host computer. For example:

```
set pka_0 container = "\\.\Tape0"
```

CHARON-VAX/XM/XL supports direct SCSI addressing of tapes.

If you have a SCSI tape and if the tape does not have a Windows driver, or the driver is not working correctly, you may instruct CHARON-VAX to address the tape directly. Before you try this method **disable** any Windows driver associated with this tape unit.

Attach the SCSI tape unit to the emulated SCSI controller as shown in the following example.

```
load physical_scsi_device tapeb1 scsi_bus=pkb
scsi_id=1
```

set tapeb1 container="\\.\scsi<scsi-port>:<scsi-bus>:<scsi-id>:<lun>”

Where:

<*scsi-port*> stands for the number assigned by the Windows operating system to the controller to which the tape drive is connected,

<*scsi-bus*> stands for the number of the SCSI bus to which the tape drive is connected (usually 0; in general, it depends on the type of SCSI controller),

<*scsi-id*> stands for SCSI ID of the tape drive.

<lun> stands for logical unit number. Can be omitted if 0.

The SCSI port can be found in registry: HKEY_Local_Machine\Hardware\Devicemap\Scsi key. The bus and id can be determined from “Start Settings / Control Panel / Administrative Tools / Computer Management / Device Manager / SCSI tape device”. Note the properties of the SCSI device that represent your tape. If not specified the LUN number defaults to zero.

The SCSI Check utility can be used to identify the correct device string used in the CHARON-VAX configuration file. See the “Utilities” chapter.

Example:

load physical_scsi_device tapeb1 scsi_bus=pkb scsi_id=1

Set tapeb1 container="\\.\scsi2:0:4”

3.6.3 atapi_scsi_device

CHARON-VAX/XM/XL supports IDE and SCSI CD-ROM drives on the host system for the initial installation of the VAX operating software from a CD-ROM distribution kit. The device location is specified in the configuration file using full device name.

Use **atapi_scsi_device** for mapping IDE and SCSI CD-ROM to CHARON-VAX/XM/XL.

atapi_scsi_device has the following parameter**:**

Set parameters	Type	Value
container[LUN]	Character	A string containing IDE or SCSI CD-ROM to map to the emulator. LUN stands for logical unit number. It must begin from 0 and has no gaps. If there is only one LUN it can be omitted.

Example:

```
load virtual_scsi_disk pka_0 scsi_bus=pka scsi_id=0
set pka_0 container="\\.\CdRom0"
```

This example assigns the first CD-ROM drive installed on your host system (CdRom0) as a SCSI device **pka_0**.

3.6.4 floppy_scsi_device

It is possible to attach one of the SCSI devices in the configuration file to the 3.5" floppy drive of the host computer. The attached floppy drive appears as RX26 SCSI floppy drive in OpenVMS/VAX running on CHARON-VAX/XM/XL. The following configuration line shows how to attach floppy drives to the SCSI controller of the emulator as DKB600:

```
load floppy_scsi_device pkb_6 scsi_bus=pkb
scsi_id=6
```

CHARON-VAX/XM/XL will make any installed floppy drives available to the emulated VAX as emulated SCSI devices.

3.7 The Ethernet adapter

To configure CHARON-VAX/XM/XL for networking follow the steps described in the section on Network installation. Here we describe the tuning parameters for the SGEC emulator and the ndis5_packet_port components.

The network subsystem is built from three parts.

- The CHARON-VAX Second Generation Ethernet Controller (SGEC) adapter.
- The packet port and
- The CHARON Packet Port Driver.

The CHARON-VAX 3100/96/98 automatically loads the SGEC. To connect the CHARON-VAX SGEC adapter instance to the external Ethernet network, load and configure the ndis5_packet_port component. This provides a bridge between the emulated SGEC adapter (eza) and the CHARON Packet Port Driver that in turn communicates with the NIC driver installed on the host system.

Use the **load** command to create an instance of the packet port (bridge):

Load parameter	Type	Value

<Instance type>	Identifier	**ndis5_packet_port** – normally used for networking under CHARON-VAX/XM/XL
Interface	String	1. Interface name.

The VAX 3100 network interfaces can be loaded as follows:

```
load <instance type> <instance name>
interface="<network interface name>"
```

You have the following options for interface name:

NIC device name.

You can obtain it from the desired connection properties. Start -> Settings -> Network and Dial-up connection; Select the connection and open its properties dialog. Use the device name specified in the 'connect using:' edit box.

Example:
"Cabletron DE500B PCI Fast Ethernet Adapter #4"

W32 device name.

Old style, ndis5 packet port instance name in the W32 device name space. Kept for compatibility.

Example:
"\Device\Packet_{322E13A2-9D9D-407C-828C-778DB3136F5C}"

Device ID.

Same as above but in short form.

Example:
"{322E13A2-9D9D-407C-828C-778DB3136F5C}"

Dos Device Name.

Device name mapped (linked) to the DOS namespace (using DefineDosDevice()). E.g. "\\.\CharonEthernet1"

Example:

```
load ndis5_packet_port eza_0 interface="Cabletron
DE500B PCI Fast Ethernet Adapter #4"
```

The following ndis5_packet_port parameters can also be specified with **set** command:

Set parameters	Type	Value
port_enable_mac_addr_change	Boolean	If **true** is specified CHARON-VAX/XM/XL sets the appropriate Ethernet address automatically. If **false** is specified set the Ethernet address manually with the NETSETUP utility. The default value is **true**.
port_ignore_on_rx	Numeric	port_ignore_on_rx provides the ability to shutdown the port when the sequential errors “on receive” exceeds the specified number. Typically errors on receive signals serious (unrecoverable) errors. By default, the value is set to the value of the port_pending_rx_number parameter. Value of ‘0’ means infinite.
port_retry_on_tx	Numeric	port_retry_on_tx controls the number of times the port attempt to transmit the packet until giving up. By default, the value is 3. Increasing this value might introduce problems in carrier loosing logic, because not all NIC drivers support carrier status query. Typically, you don’t need to increase the value.

port_pending_rx_number	Numeric	port_pending_rx_number sets the number of pending receive buffers. The default value is 63. The maximum value allowed by Charon's Protocol Driver, is 195. You may want to increase the port_pending_rx_number when you have very busy networking and experience problems like loosing connections not related to the carrier loss. Typically, you don't need to change this parameter.
port_pending_tx_number	Numeric	port_pending_tx_number sets the number of buffers the port uses for transmit. The default value is 62. You may want to increase the port_pending_tx_number value if the log file indicates dropped TX packets due to TX queue overflow. Typically, you don't need to change this parameter.

Example:

```
set eza_0 port_enable_mac_addr_change=false
```

Once you have loaded one of the packet interfaces connect the built-in Ethernet Controller EZA to this port in the following way:

`set eza interface=<instance name>,` example**:**

```
set eza interface=eza_0
```

The built-in SGEC controller emulator (eza) has the following additional parameters that you may specify with the set command:

Load parameter	Type	Value
Station_address	Ethernet address	Station_address provides the user with the ability to configure the adapter's permanent address. By default, adapter's permanent address is read from the host system's NIC. Setting the station_address required when you need to configure the satellite (remotely booted) system which will run DECnet or when software you want to migrate under emulator uses network adapter permanent address for license protection. Format: XX-XX-XX-XX-XX-XX Or XX:XX:XX:XX:XX:XX
rx_fifo_size	Numeric	Rx_fifo_size sets the receive FIFO size. The value is specified in Kb and by default is pre-calculated from the connected port's size of receive queue. Typically, you don't need to change the rx_fifo_size parameter. It is available for extended tuning and bug hunting purposes.

Example:

```
set eza station_address="XX:XX:XX:XX:XX:XX"
```

3.8 VAX Non volatile Memory (NVRAM)

Set parameter	Type	Value
container	String	This parameter tells the emulator where to preserve NVRAM content. It will keep the current time of the emulated VAX (when you do not run the emulator) and some (but not all!) of the console parameters.

The NVRAM container can be specified as follows:

`set toy <parameter>=<Value>`, for example:

`set toy container = “charon.dat”`

In the above, TOY is an abbreviation for TIME-OF-YEAR. If your VAX operating system cannot determine the date and time automatically the NVRAM file is probably corrupted. Delete the file; reboot your system, set the date and time. Shut down the operating system and close CHARON-VAX. Start up again and check that the emulator can determine the date and time automatically.

The CHARON-VAX system time is stored as an offset to the host system time. Thus do not reset the host system time. Note that if you are using a date limited CHARON-VAX license re-setting the host system time can permanently damage the license.

3.9 Flash ROM state

The flash ROM stores the VAX console parameters including any saved boot settings.

Set parameter	Type	Value

container	String	This parameter tells the emulator where to store intermediate state of the Flash ROM. It will keep the rest of console parameters.

The log file of the emulated VAX 3100 can be set as follows:

`set rom <parameter>=<Value>`, for example:

`set rom container = “charon.rom”`

If your operating system cannot determine the boot parameters the Flash ROM file is probably corrupted, delete the file; reboot your system set the boot parameters and check that the system boots correctly. Shut down the operating system and close CHARON-VAX. Start up again and check that the emulator can determine the boot parameters automatically.

3.10 The RQDX3 MSCP (disk) Controller (Q-bus)

The CHARON-VAX/XM/XL QBUS systems provide support for RQDX3 disk controller. The original RQDX3 disk controller is capable of serving up to 4 disk units. But the CHARON-VAX/XM/XL extends this limit so that the RQDX3 disk controller can be configured with up to 256 disk units.

Use the following command to load an instance of RQDX3 disk controller:

`LOAD RQDX3 <logical name>`

`Example`

`load RQDX3 DUA`

The RQDX3 offers the following configuration parameters which can be specified with SET command:

Set parameter	Type	Value
Address	Numeric	Specifies the CSR address. The address must be a valid QBUS 22-bit wide address in IO space. Initial value is 017772150 which is the factory setting for RQDX3 disk controller.
max_n_of_units	Numeric	Specifies the maximum number of units supported by the controller. Possible values are 4…256. Initially it is set to 256.
container[N] **N=0…(max_n_units-1)**	String	Specifies location of disk container. It can be either name of .VDISK (.DSK) file or name of physical disk: "\\.\PhysicalDriveX" – for local fixed disks (IDE, SCSI, SATA, …); "\\.\CdRomX" or "\\.\<Letter>:" - for CD-ROM, DVD drives (IDE, SCSI, …); "\\.\A:" or \\.\B: or whatever – for floppy drives (1.2, 1.44, 2.88MB);

The ADDRESS parameter allow loading of several instances of RQDX3. The ADDRESS parameter value must be unique for every instance of RQDX3. For example:

```
load RQDX3 DUA address=017772150 max_n_of_units=4
load RQDX3 DUB address=017760334 max_n_of_units=4
```

Note that the RQDX3 disk controller available as part of DEMO distribution is restricted to 4 disk units, and the MAX_N_OF_UNITS parameter is not available.

3.11 The TQK50 TMSCP (tape) Controller (Q-bus)

The CHARON-VAX/XM/XL QBUS systems provide support for TQK50 tape controller. The original TQK50 tape controller is capable of serving only 1 tape unit. But the CHARON-VAX/XM/XL extends this limit so that the TQK50 tape controller can be configured with up to 256 tape units.

Use the following command to load an instance of TQK50 tape controller:

```
LOAD TQK50 <logical name>
```

```
Example
```

```
load TQK50 MUA
```

The TQK50 offers the following configuration parameters which can be specified with SET command:

Set parameter	Type	Value
address	Numeric	Specifies the CSR address. The address must be a valid QBUS 22-bit wide address in IO space. Initial value is 017774500 which is the factory setting for TQK50 tape controller.
container[N] **N=0…255**	String	Specifies location of tape container. It can be either name of .VTAPE (.MTD) file or name of physical tape drive: "\\.\TapeX" – for local physical tape drive (SCSI);

The ADDRESS parameter allow loading of several instances of TQK50. The ADDRESS parameter value must be unique for every instance of TQK50. For example:

```
load TQK50 MUA address=017774500
load TQK50 MUB address=017774504
```

3.12 The DHV11 serial line controller (Q-bus)

The CHARON-VAX/XM/XL QBUS systems provide support for DHV11 asynchronous serial line multiplexer. The DHV11 asynchronous serial line multiplexer is capable of serving up to 8 asynchronous serial lines.

Use the following command to load an instance of DHV11 asynchronous serial line multiplexer:

```
LOAD DHV11 <logical name>
Example
load DHV11 TXA
```

The DHV11 offers the following configuration parameters which can be specified with SET command:

Set parameter	Type	Value
Address	Numeric	Specifies the CSR address. The address must be a valid QBUS 22-bit wide address in IO space. Initial value is 017760440 which is the factory setting for DHV11 asynchronous serial line multiplexer.
Vector	Numeric	Specifies the interrupt vector. Initial value is 0300 which is the factory setting for DHV11 asynchronous serial line multiplexer.
line[N] **N=0…7**	Identifier	Specifies name of serial_line object in the configuration to which the *N*th line of the multiplexer is connected.

communication[N] **N=0…7**	String	Specifies type of communication over the Nth line of the multiplexer. Two values are supported: “ASCII” and “BINARY”. Default value is “ASCII”.
rts[N] **N=0…7**	String	Controls RTS (Request To Send) signal of the *N*th line of the multiplexer. “ON” - assert the RTS signal; “OFF” – clear the RTS signal; “DTR” - assert the RTS signal as soon as the DTR signal is asserted; When left blank (initial state), the level of RTS signal is as requested by VAX software.
tx_q_max_depth[N] **N=0…7**	Numeric	Specifies depth of the TX FIFO for the *N*th line of the multiplexer. Possible values are 1…1000, initially it is set to 1 to properly represent the hardware limitation of DHV11 multiplexer. Values greater then 1 do improve transmission rate of the corresponding line, but break correspondence to the original hardware.

The ADDRESS and VECTOR parameters allow loading of several instances of DHV11. Both ADDRESS and VECTOR parameters values must be unique for every instance of DHV11. For example:

```
load DHV11 TXA address=017760440 vector=0300
load DHV11 TXB address=017760460 vector=0310
```

3.13 The DHQ11 serial line controller (Q-bus)

The CHARON-VAX/XM/XL Q-bus systems provide support for DHQ11 asynchronous serial line multiplexer. The DHQ11 asynchronous serial line multiplexer is capable of serving up to 8 asynchronous serial lines.

Use the following command to load an instance of DHQ11 asynchronous serial line multiplexer:

`LOAD DHQ11/DHV11 <logical name>`

`Example`

`load DHQ11/DHV11 TXA`

The DHQ11 offers the following configuration parameters which can be specified with SET command:

Set parameter	Type	Value
Address	Numeric	Specifies the CSR address. The address must be a valid QBUS 22-bit wide address in IO space. Initial value is 017760440 which is the factory setting for DHQ11 asynchronous serial line multiplexer.
Vector	Numeric	Specifies the interrupt vector. Initial value is 0300 which is the factory setting for DHQ11 asynchronous serial line multiplexer.
line[N] **N=0…7**	Identifier	Specifies name of serial_line object in the configuration to which the *N*th line of the multiplexer is connected.

communication[N] **N=0…7**	String	Specifies type of communication over the Nth line of the multiplexer. Two values are supported: “ASCII” and “BINARY”. Default value is “ASCII”.
rts[N] **N=0…7**	String	Controls RTS (Request To Send) signal of the *N*th line of the multiplexer. “ON” - assert the RTS signal; “OFF” – clear the RTS signal; “DTR” - assert the RTS signal as soon as the DTR signal is asserted; When left blank (initial state), the level of RTS signal is as requested by VAX software.

The ADDRESS and VECTOR parameters allow loading of several instances of DHQ11. Both ADDRESS and VECTOR parameters values must be unique for every instance of DHQ11. For example:

```
load DHQ11/DHV11 TXA address=017760440 vector=0300
load DHQ11/DHV11 TXB address=017760460 vector=0310
```

3.14 The CXY08 serial line controller (Q-bus)

The CHARON-VAX/XM/XL Q-bus systems provide support for CXY08 asynchronous serial line multiplexer. The CXY08 asynchronous serial line multiplexer is capable of serving up to 8 asynchronous serial lines.

Use the following command to load an instance of CXY08 asynchronous serial line multiplexer:

```
LOAD CXY08/DHV11 <logical name>
Example
```

load CXY08/DHV11 TXA

The CXY08 offers the following configuration parameters which can be specified with SET command:

Set parameter	Type	Value
Address	Numeric	Specifies the CSR address. The address must be a valid QBUS 22-bit wide address in IO space. Initial value is 017760440 which is the factory setting for CXY08 asynchronous serial line multiplexer.
Vector	Numeric	Specifies the interrupt vector. Initial value is 0300 which is the factory setting for CXY08 asynchronous serial line multiplexer.
line[N] **N=0…7**	Identifier	Specifies name of serial_line object in the configuration to which the *N*th line of the multiplexer is connected.
communication[N] **N=0…7**	String	Specifies type of communication over the *N*th line of the multiplexer. Two values are supported: “ASCII” and “BINARY”. Default value is “ASCII”.
rts[N] **N=0…7**	String	Controls RTS (Request To Send) signal of the *N*th line of the multiplexer. “ON” - assert the RTS signal; “OFF” – clear the RTS signal; “DTR” - assert the RTS signal as soon as the DTR signal is asserted; When left blank (initial state), the level of RTS signal is as requested by VAX software.

The ADDRESS and VECTOR parameters allow loading of several instances of CXY08. Both ADDRESS and VECTOR parameters values must be unique for every instance of CXY08. For example:

```
load CXY08/DHV11 TXA address=017760440 vector=0300
load CXY08/DHV11 TXB address=017760460 vector=0310
```

3.15 The CXA16 serial line controller (Q-bus)

The CHARON-VAX/XM/XL Q-bus systems provide support for CXA16 asynchronous serial line multiplexer. The CXA16 asynchronous serial line multiplexer is capable of serving up to 16 asynchronous serial lines.

Use the following command to load an instance of CXA16 asynchronous serial line multiplexer:

```
LOAD CXA16/DHV11 <logical name>
```

Example

```
load CXA16/DHV11 TXA
```

The CXA16 offers the following configuration parameters which can be specified with SET command:

Set parameter	Type	Value
address	Numeric	Specifies the CSR address. The address must be a valid QBUS 22-bit wide address in IO space. Initial value is 017760440 which is the factory setting for CXA16 asynchronous serial line multiplexer.
vector	Numeric	Specifies the interrupt vector. Initial value is 0300 which is the factory setting for CXA16 asynchronous serial line multiplexer.

line[N] **N=0…15**	Identifier	Specifies name of serial_line object in the configuration to which the *N*th line of the multiplexer is connected.
communicatio n[N] **N=0…15**	String	Specifies type of communication over the *N*th line of the multiplexer. Two values are supported: “ASCII” and “BINARY”. Default value is “ASCII”.
rts[N] **N=0…15**	String	Controls RTS (Request To Send) signal of the *N*th line of the multiplexer. “ON” - assert the RTS signal; “OFF” – clear the RTS signal; “DTR” - assert the RTS signal as soon as the DTR signal is asserted; When left blank (initial state), the level of RTS signal is as requested by VAX software.

The ADDRESS and VECTOR parameters allow loading of several instances of CXA16. Both ADDRESS and VECTOR parameters values must be unique for every instance of CXA16. For example:

```
load CXA16/DHV11 TXA address=017760440 vector=0300
load CXA16/DHV11 TXB address=017760460 vector=0310
```

The CHARON-VAX/XM/XL emulator also supports CXB16 asynchronous serial line multiplexer.

3.16 The DEQNA Ethernet Controller (Q-bus)

The CHARON-VAX/XM/XL Q-bus systems provide support for DEQNA Ethernet controller.

Use the following command to load an instance of DEQNA Ethernet controller:

```
LOAD DEQNA <logical name>
```

Example

```
load DEQNA XQA
```

The DEQNA offers the following configuration parameters which can be specified with SET command:

Set parameter	Type	Value
address	Numeric	Specifies the CSR address. The address must be a valid QBUS 22-bit wide address in IO space. Initial value is 017774440 which is the factory setting for DEQNA Ethernet controller.
rx_fifo_size	Numeric	The RX_FIFO_SIZE sets the receive FIFO size. The value is specified in KB and by default is pre-calculated from the connected port's size of receive queue. Typically, you don't need to change the RX_FIFO_SIZE parameter. It is available for extended tuning and bug hunting purposes.

station_address	Ethernet address	The STATION_ADDRESS provides the user with the ability to configure the adapter's permanent address. By default, adapter's permanent address is read from the host system's NIC (provided that the adapter is connected to host NIC). Setting the STATION_ADDRESS required when you need to configure the satellite (remotely booted) system which will run DECnet or when software you want to migrate under emulator uses network adapter permanent address for license protection. Format: XX-XX-XX-XX-XX-XX or XX:XX:XX:XX:XX:XX where XX represents 8-bit value in hexadecimal form.

The ADDRESS parameter allow loading of several instances of DEQNA. The ADDRESS parameter value must be unique for every instance of DEQNA. For example:

```
load DEQNA XQA address=017774440
load DEQNA XQB address=017764460
```

3.17 The DELQA Ethernet Controller (Q-bus)

The CHARON-VAX/XM/XL Q-bus systems provide support for DELQA Ethernet controller.

Use the following command to load an instance of DELQA Ethernet controller:

LOAD DELQA/DEQNA <logical name>

Example

load DELQA/DEQNA XQA

The DELQA offers the following configuration parameters which can be specified with SET command:

Set parameter	Type	Value
address	Numeric	Specifies the CSR address. The address must be a valid QBUS 22-bit wide address in IO space. Initial value is 017774440 which is the factory setting for DELQA Ethernet controller.
rx_fifo_size	Numeric	The RX_FIFO_SIZE sets the receive FIFO size. The value is specified in KB and by default is pre-calculated from the connected port's size of receive queue. Typically, you don't need to change the RX_FIFO_SIZE parameter. It is available for extended tuning and bug hunting purposes.

station_address	Ethernet address	The STATION_ADDRESS provides the user with the ability to configure the adapter's permanent address. By default, adapter's permanent address is read from the host system's NIC (provided that the adapter is connected to host NIC). Setting the STATION_ADDRESS required when you need to configure the satellite (remotely booted) system which will run DECnet or when software you want to migrate under emulator uses network adapter permanent address for license protection. Format: XX-XX-XX-XX-XX-XX or XX:XX:XX:XX:XX:XX where XX represents 8-bit value in hexadecimal form.

The ADDRESS parameter allow loading of several instances of DELQA. The ADDRESS parameter value must be unique for every instance of DELQA. For example:

```
load DELQA/DEQNA XQA address=017774440
load DELQA/DEQNA XQB address=017764460
```

3.18 The DESQA Ethernet Controller (Q-bus)

The CHARON-VAX/XM/XL Q-bus systems provide support for DESQA Ethernet controller.

Use the following command to load an instance of DESQA Ethernet controller:

LOAD DESQA/DEQNA <logical name>

Example

load DESQA/DEQNA XQA

The DESQA offers the following configuration parameters which can be specified with SET command:

Set parameter	Type	Value
address	Numeric	Specifies the CSR address. The address must be a valid QBUS 22-bit wide address in IO space. Initial value is 017774440 which is the factory setting for DESQA Ethernet controller.
rx_fifo_size	Numeric	The RX_FIFO_SIZE sets the receive FIFO size. The value is specified in KB and by default is pre-calculated from the connected port's size of receive queue. Typically, you don't need to change the RX_FIFO_SIZE parameter. It is available for extended tuning and bug hunting purposes.

station_address	Ethernet address	The STATION_ADDRESS provides the user with the ability to configure the adapter's permanent address. By default, adapter's permanent address is read from the host system's NIC (provided that the adapter is connected to host NIC). Setting the STATION_ADDRESS required when you need to configure the satellite (remotely booted) system which will run DECnet or when software you want to migrate under emulator uses network adapter permanent address for license protection. Format: XX-XX-XX-XX-XX-XX or XX:XX:XX:XX:XX:XX where XX represents 8-bit value in hexadecimal form.

The ADDRESS parameter allow loading of several instances of DESQA. The ADDRESS parameter value must be unique for every instance of DESQA. For example:

```
load DESQA/DEQNA XQA address=017774440
load DESQA/DEQNA XQB address=017764460
```

3.19 The LPV11 Parallel Line printer Controller (Q-bus)

The CHARON-VAX/XM/XL Q-bus systems provide support for LPV11 Parallel Line Printer controller. The LPV11 supported is implemented using CHAPI, therefore configuration guidelines for LP11 differ from those of non-CHAPI components.

Use the following command to load an instance of LPV11 Parallel Line Printer controller:

`LOAD CHAPI <logical name> DLL=LPV11`

`Example`

`load CHAPI LPA0 dll=LPV11`

For more information on configuring and use of CHAPI components please refer to *<proper document name>.*

The LPV11 (as it is a CHAPI component) offers the following configuration parameters which can be specified with SET command:

Set parameter	Type	Value
address	Numeric	Specifies the CSR address. The address must be a valid QBUS 22-bit wide address in IO space. Initial value is 017777514 which is the factory setting for LPV11 Parallel Line Printer controller.
vector	Numeric	Specifies the interrupt vector. Initial value is 0200 which is the factory setting for LPV11 Parallel Line Printer controller.
parameters	String	Specifies name of file to which the LPV11 prints the text or name of physical printer: "\\.\LPT1:" - to connect to physical printer;

Chapter 4 Creating a VAX configuration

4.1 Creating a VAX configuration manually.

This chapter explains step by step how to create a simple configuration file for CHARON-VAX/XM/XL. For this example the following configuration will be created:

- 256Mb of memory
- DIT disabled
- Console emulated by Hyperterm
- One disk emulated by a disk image
- One Floppy disk mapped on A:
- CD-ROM drive
- Ethernet Adapter

See the in the Chapter on Configuration and the configuration examples in the Appendix on for more complete explanations.

4.1.1 Define the hardware model

First set the model of VAX you wish to run with a line like this:

set session hw_model="MicroVAX_3100_Model_98"

4.1.2 Log, NVRAM and ROM files

Now configure files for the system log, the NVRAM and the ROM file with the following lines:

```
set session log="charon.log"
set toy container="charon.dat"
set rom container="charon.rom"
```

By default these files are stored in the folder from where CHARON-VAX is executed. If you would like you can specify some exact path for your log file, for instance:

```
set session log="C:\Charon\Logs\charon.log"
```

4.1.3 DIT option

Now specify whether Dynamic Instruction Translation (DIT) will be enabled. Please note that only the Plus version of the emulator allows DIT. By default the system configures DIT as defined by your license key. To see the effect of running the emulator without DIT when you do have a Plus license code as follows:

```
set cpu dit_enable=false
```

To enable your Plus (DIT) option specify “true” instead of “false”.

4.1.4 Amount of memory

Set the required amount of memory (in our example it’s 256Mb) with the following line:

```
set ram size=256
```

Note that your license key restricts the maximum emulated memory, e.g. CHARON-VAX/XM is limited to 128MB and CHARON-VAX/XL is limited to 512MB of memory.

4.1.5 Default console

The default console will be connected to 4th port (0-3) of the QUART adapter and emulated by CHARON-VAX/XM/XL built-in terminal emulator:

```
load virtual_serial_line OPA0 port=10003 application="opa0.ht"
set quart line[3]=OPA0
```

4.1.6 Disk drive

Now configure a disk image on the first built-in SCSI adapter named PKA. The address of this disk will be 0, so it will appear in the VMS environment as DKA0:

```
load virtual_scsi_disk pka_0 scsi_bus=pka scsi_id=0
set pka_0 container[0]="system.vdisk”
```

Please note that the emulator will look for the disk image in directory “VAX3100\bin” under your CHARON-VAX/XM/XL installation directory if you use this syntax. It may be better to specify the exact path to your disk image as in this example:

```
set pka_0
container[0]="C:\Charon\Disks\system.vdisk”
```

Also note that in this example the container [0] is used. As zero is the default logical unit number (LUN) the following, less complete, format has the same effect.

```
set pka_0 container="C:\Charon\Disks\system.vdisk"
```

4.1.7 Floppy disk

To configure your A: floppy disk for the emulator as, for instance DKA100: (PKA adapter, SCSI address 1) write the following in your configuration file:

```
load floppy_scsi_device pka_1 scsi_bus=pka scsi_id=1
```

4.1.8 CD-ROM drive

To configure your default CD-ROM drive (\\.\Cdrom0) for the emulator as, for instance DKA200: (PKA adapter, SCSI address 2) code as follows:

```
load atapi_scsi_device pka_2 scsi_bus=pka scsi_id=2
set pka_2 container="\\.\CdRom0"
```

4.1.9 Ethernet Adapter

The last step is configuring your emulated Ethernet adapter. Make sure that you have already installed “Charon packet driver (NDIS5)” and configured it properly. If not please refer to the chapter on Network Installation.

Open up the NetSetup utility (see the Utilities chapter for details) from “Start->Programs->CHARON Emulators->Utilities”, choose the desired network card to be used with CHARON-VAX/XM/XL and copy its interface name by selecting and pressing Ctrl-C. In our example let’s suppose that this name is “\Device\Packet_{503495A6-5C2A-461F-A08F-E6E2B5A8A20A}”

Write the following in your configuration file:

```
load ndis5_packet_port eza_0
interface="\Device\Packet_{503495A6-5C2A-461F-A08F-E6E2B5A8A20A}"
```

```
and connect the packet port to the built-in SGEC as follows
```

```
set eza interface=eza_0
```

Remember to specify your own interface name from NetSetup utility instead the one given in this example.

Your resulting configuration file may look like this:

```
#The CHARON-VAX model

set session hw_model="MicroVAX_3100_Model_98"

#Logs and system parameters
set toy container="charon.dat"
set rom container="charon.rom"
set session log="C:\Charon\Logs\charon.log"
set cpu dit_enable=false
set ram size=256

#CONSOLE
load virtual_serial_line OPA0 port=10003 application=”opa0.ht”
set quart line[3]=OPA0
```

```
#DISK
load virtual_scsi_disk pka_0 scsi_bus=pka scsi_id=0
set pka_0 container="C:\Charon\Disks\system.vdisk"

#FLOPPY DISK
load floppy_scsi_device pka_1 scsi_bus=pka scsi_id=1

#ATAPI CD
load atapi_scsi_device pka_2 scsi_bus=pka scsi_id=2
set pka_2 container="\\.\CdRom0"

#Network
load ndis5_packet_port eza_0
interface="\Device\Packet_{503495A6-5C2A-461F-A08F-
E6E2B5A8A20A}"
set eza interface=eza_0
```

4.2 Starting CHARON-VAX with this configuration

Save your configuration file under CHARON-VAX/XM/XL installation directory in subdirectory VAX3100\bin. If you name your configuration file charon.cfg you may now start CHARON-VAX using the desktop or start menu short cuts, if available. If for example you name the configuration file “mycfg.cfg” you may create your own shortcut or start CHARON-VAX with this configuration file from the CHARON Launcher. The CHARON Launcher is a useful tool to be used when debugging a configuration file as you can see any errors and the log file from the same Window from which you can also edit and start your configuration. Alternatively open up a DOS command console, set the directory to VAX3100\bin and issue the following command:
>charon mycfg.cfg

4.3 Using the CHARON configuration Wizard

This chapter explains how to create the configuration described in the above chapter using the CCW (CHARON Configuration Wizard) utility.

Start the utility and answer the requests as follows (bold font):

```
If you have installed all required drivers and devices
then type Y to continue
```

```
Continue? [Y]: <ENTER>
Enter configuration file [ C:\Program Files\CHARON
Emulators\VAX3100\Bin\charon.c
fg ]: C:\Program Files\CHARON Emulators\VAX3100\Bin\mycfg.cfg
Choose Desired Hardware Model:

     1. MicroVAX_3100_Model_98
     2. MicroVAX_3100_Model_96

Enter your choice [1]: <ENTER>
Do you want to enable DIT (recommended)? [Y]:N
Enter desired amount of emulated memory in Mb [160]:256
How many consoles to configure? [ 1, maximum: 4 ]:
<ENTER>
Choose the options for the Console:

     1. Connect a terminal emulator on your host
computer
     2. Connect from a terminal emulator on a remote
computer
     3. Connect a real terminal/device to a COM port

Enter your choice [1]: <ENTER>
Note: Port 10003 will be used for this line
Do you want the terminal application started
automatically? [Y]: <ENTER>
Enter the path to the terminal application or its
configuration file [ opa0.ht ]
: <ENTER>
How many disks to configure? [ 1, maximum: 14 ]: <ENTER>
Do you want to use physical disks? [Y]:N
Configuring disk 1:
Enter full path to disk image [ C:\Disks\disk1.vdisk ]:
C:\Charon\Disks\system.vdisk
Do you want to use host CD-ROM drive? [Y]: <ENTER>
Do you want to use host floppy drive? [Y]: <ENTER>
Do you want to have networking? [Y]: <ENTER>

The configuration file has been generated. Press any key
to exit.
```

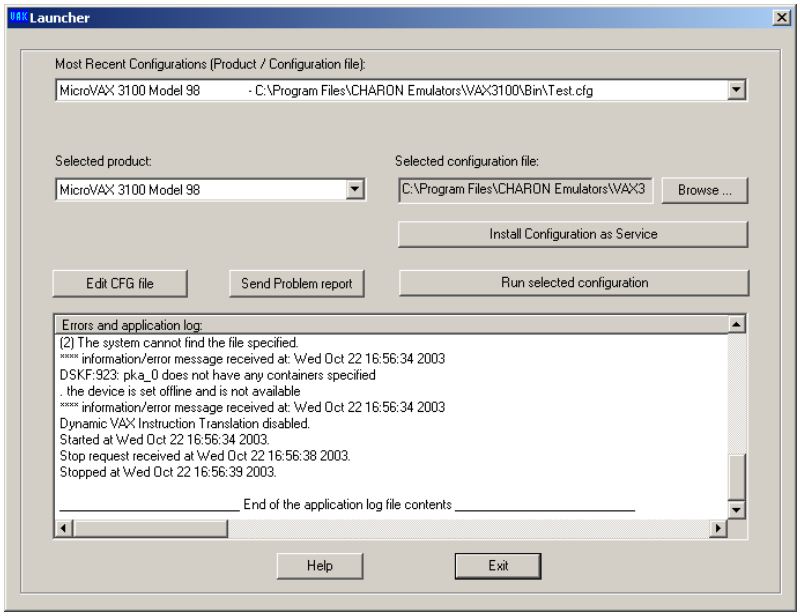
As a result the following configuration file will be generated:

```
set session hw_model="MicroVAX_3100_Model_98"


set session log="charon.log"

set toy container="charon.dat"

set rom container="charon.rom"

set cpu dit_enable=false

set ram size=256

load virtual_serial_line OPA0 port=10003 application="opa0.ht"

set quart line[3]=OPA0

load virtual_scsi_disk pka_0 scsi_bus=pka scsi_id=0

set pka_0 container="C:\Charon\Disks\system.vdisk"

load atapi_scsi_device pka_1 scsi_bus=pka scsi_id=1
container="\\.\CdRom0"

load floppy_scsi_device pka_2 scsi_bus=pka scsi_id=2

load ndis5_packet_port eza_0
interface="\Device\Packet_{A906270D-9CFF-45F0-839B-
F2918846AB75}"

set eza interface=eza_0
```

Chapter 5 Utilities

5.1 CHARON Launcher

Use the CHARON Launcher . to start CHARON manually. Click on Help or type F1 for help. Select the CHARON product and the configuration file you require to run. Either select the combination from the list of Most Recent Configurations or select the product and configuration manually.



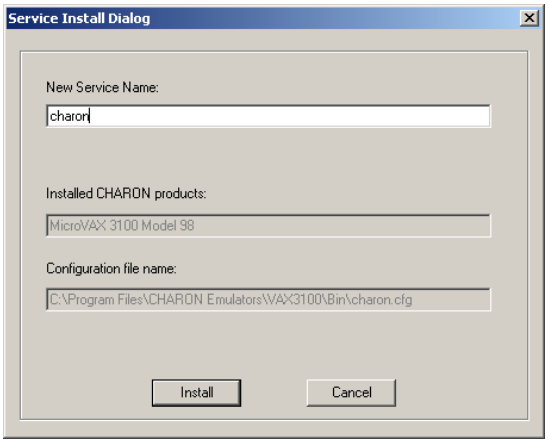
Click “Run the selected configuration file” and CHARON will start. Any errors and the application log will be displayed in the Launcher Window immediately after CHARON starts. While CHARON is running, the Launcher updates the log file contents each 60 seconds.

After CHARON stops the Launcher loads the full application log contents for review.

The CHARON Launcher can edit the selected configuration file using the notepad editor. The combination of running the selected configuration, displaying errors and the log and editing the configuration makes the CHARON Launcher very useful for debugging new installations/configuration files or examining error conditions.

Send a problem report straight to your support team by clicking on the “send problem report” button. Complete the message by filling in the blanks in the draft message generated by CHARON.

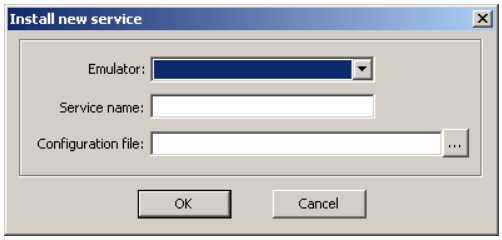
You can install CHARON-VAX/XM/XL as service by pressing the button “Install Configuration as service”. The following dialog will appear:



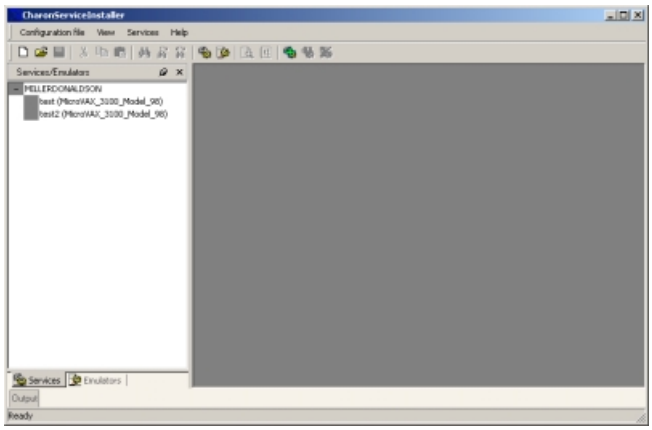
Enter the desired service name in “New Service Name” and press the “Install” button.

5.2 CHARON Service Installer

The CHARON Service Installer utility installs a selected CHARON-VAX/XM/XL configuration as Windows system service. In this mode there is no user login required to start CHARON-VAX. See the utilities menu. Only install a configuration as a service after running it without problems manually from the CHARON Launcher or a short cut. Define the service to be installed by selecting the product to run, the name you wish to give this service and the configuration file.

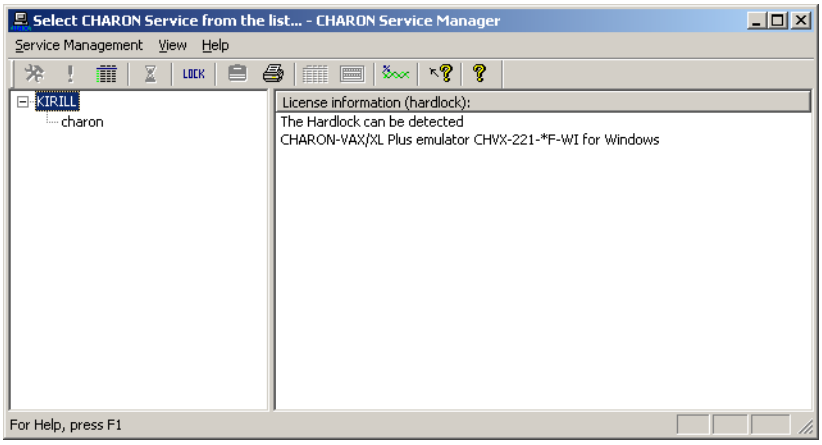


The Service installer main Window is shown below. This Window may be used to view what CHARON products and services are installed. For a given service you may view the Configuration (F7) or the log file (F8). You may create and edit configuration files from the Service Installer. You may also add verify or delete services.

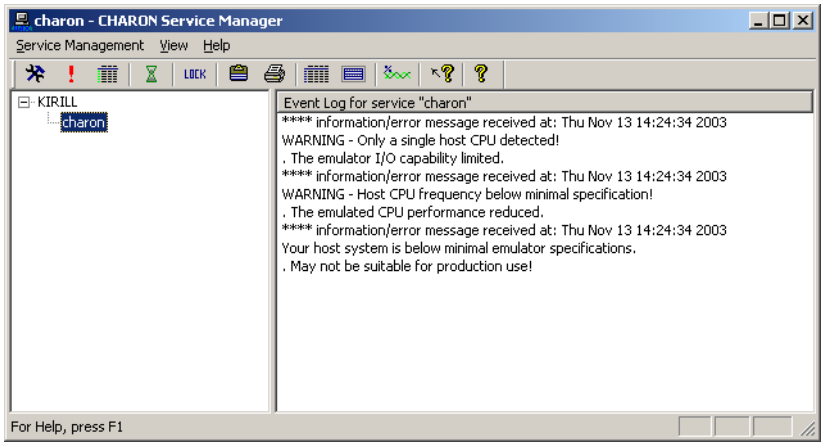


5.3 CHARON Service Manager

Use the CHARON Service Manager to manage CHARON services installed and running on your computer. Run the utility from the CHARON start menu. Click Help or hit F1 for help. Below is the utility interface:

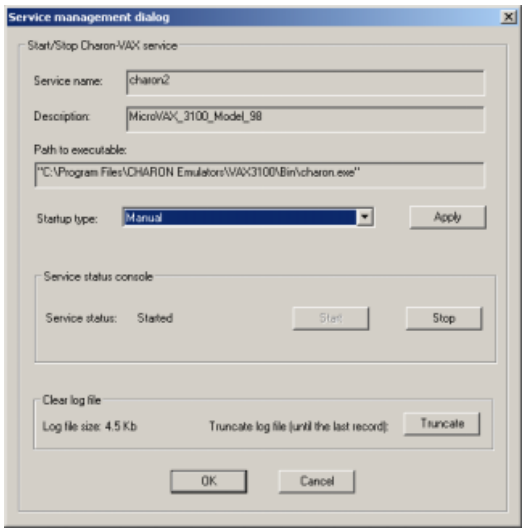


The tree structure on the left presents the CHARON-VAX/XM/XL services installed on the computer. The right hand pane displays Hardlock information. Clicking on a service name leads to an event log display in the right hand pane.



The buttons on the panel have the following meaning (left to right):

1. *Manage CHARON-VAX services* invokes the following dialog:



"Service Name" stands for name of the chosen service, "Description" displays the CHARON model, "Path to executable" refers to the executable to run.

“Startup type” can be “Manual”, “Automatic” or “Disabled”. Type or choose the desired type and press “Apply” to apply the setting.

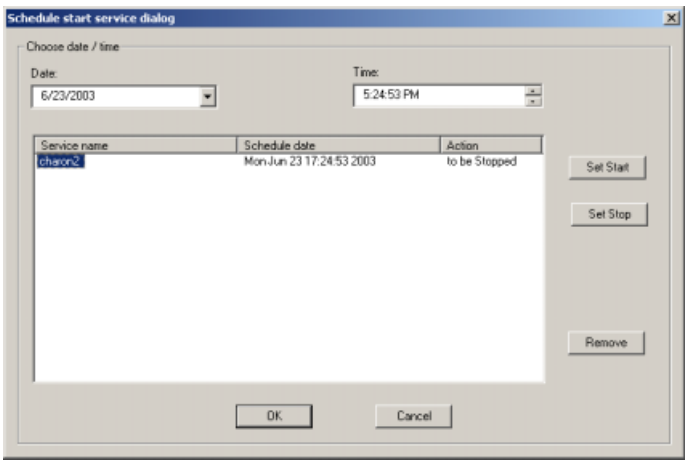
“Service Status” gives information on the current status of the service. You can start the service or stop it depending on its current status.

The last option is information on the event log file size. Use the “Truncate” button to reduce the size of unnecessarily large log file.

2. *Remove selected CHARON-VAX services* removes the service you currently selected.

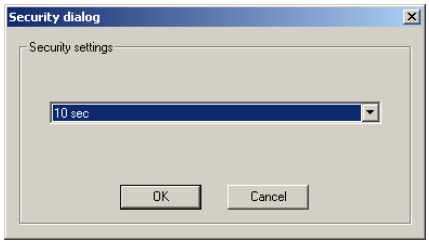
3. *Update the list of installed CHARON services* – updates the service list shown in the left pane of the application window. This button is useful if you add new services during the CHARON Service manager run time. New added services are invisible until you restart the CHARON Service manager or update this list. Also available through corresponding item in the “Service Management” submenu.

4. *Schedule start service* (also available through corresponding item in “Service Management” menu) sets the start and if necessary stop time of a selected service. It invokes the following dialogue:



Press the “Set start” or “Set Stop” buttons to apply the date and time. To remove any schedule date choose it and press the “Remove” button. Note that all scheduled tasks proceed only if the CHARON Service manager runs. No scheduled information is stored in the system registry. When CHARON service manager restarts the scheduled task queue will be empty.

5. *Security Setting* (also available through corresponding item in “Service Management” menu). This function locks the computer after a given amount of time. Enter the Windows User name/password combination to restart access. By default this function is disabled. Below is the dialog screen:

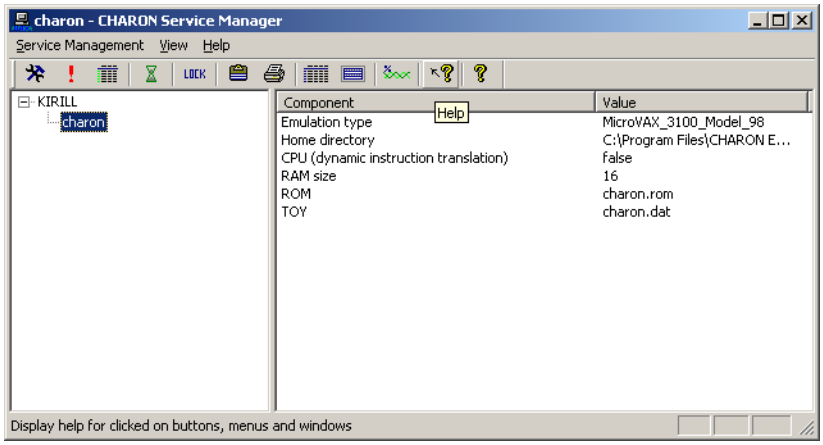


6. *Mail Problem Report* (also available through mail item in “Service Management” menu) automatically formats a problem report mail. This feature is available only if the CharonMAPI.dll is present in the application directory. If the DLL is not installed or manually removed, this button and the toolbar icon are disabled.

7. *Print the active document* opens up the standard printer dialogue to print the right hand panel of the application. Use the menu “Service Management” to customize printing through “Print Preview” and “Print Setup”.

8. *Display service startup and error log* displays service startup and event log of chosen service in right panel of the application. This function is also available in “View” menu.

9. *Display the configuration* displays the configuration of the chosen service in the right panel of the application. This function is also available in the “View” menu.



10. System Monitor invokes the Windows system monitor to trace the chosen service activity. This function is also available in the “View” menu.

11. The “question mark” buttons provide access to the help system.

5.4 Charon Configuration Wizard (CCW)

Using this utility you can quickly build a configuration file by answering simple questions. Run the utility from the Utilities entry in the CHARON start Menu. On start-up CCW checks the system configuration and license and ask whether it can proceed. Depending on the configuration of the system and the license CCW asks certain questions and provides default or recommended values. If the specified configuration file name exists it will be overwritten.

CCW is able to create and initialize disk images – so the only thing required in this case is simple specification of the disk type and file name.

After the last question CCW finishes creating the configuration file. You can quit any time from the CCW by pressing “Ctrl-C”. If CCW has created configuration file it remembers most of the answers, so you can re-run the utility at anytime to correct your configuration.

Note: CCW cannot load and edit a configuration file – it can only create a new file or overwrite an existing one.

5.5 CHARON Automated Problem Report Configuration

Complete the CHARON Automated Problem Report Configuration dialogue at installation time or access this utility (utilities menu) to update the contact information as required.

Enter the customer contact information in all cases.

Enter your CHARON-VAX distributor information (VAR) if the problem report should be sent to the VAR. If no VAR information is entered the problem report is sent to charon@softresint.com.

Enter the Software Resources International Representative details if the representative should receive a copy of the problem report.

5.6 CHARON CPU usage optimization (IDLE_VMS_PKG)

The CHARON CPU usage optimization package is a collection of utilities, namely IDLE, SHUTDOWN and SHUTDOWN_R. You will find this package in the CHARON-VAX installation directory VAX3100\bin. Unzip the IDLE_VMS_PKG.ZIP and extract the disk image. Specify this image in the CHARON-VAX configuration file, boot from your system disk and mount the disk with the following command:

$ MOUNT <device name> /OVER=ID

In the root of this disk you will see 3 executable files: IDLE.EXE, SHUTDOWN.EXE and SHUTDOWN_R.EXE.

IDLE.EXE – Reduces CHARON host CPU usage close to 0% whenever a VMS system running on CHARON-VAX is idle. You may include a call to the IDLE.EXE in the system startup file.

SHUTDOWN.EXE – Stops CHARON after 30 seconds after the utility is called. Those 30 seconds are intended to shutdown VMS running on CHARON.

SHUTDOWN3.EXE – Stops CHARON after 3 minutes after the utility is called. Those 3 minutes are intended to shutdown VMS running on CHARON.

SHUTDOWN5.EXE – Stops CHARON after 5 minutes after the utility is called. Those 5 minutes are intended to shutdown VMS running on CHARON.

SHUTDOWN_R.EXE – Cancels a pending request to SHUTDOWN.

To run any of those utilities issue the following command:

$ RUN <name of the utility>

Example:

$ RUN IDLE.EXE

Appendix A Configuration files

CHARON-VAX/XM and /XL use a configuration file, which is located by default in the directory “VAX”. Three sample configuration files are supplied.

Cheetah.cfg demonstrates the options for the MicroVAX 3100 model 96 and MicroVAX 3100 model 98

Cheetah-q.cfg demonstrates the options for the VAX 4100 model 106 and the VAX 4100 model 108

Mayfair.cfg demonstrates the options for the, MicroVAX 3600, MicroVAX 3900, VAXserver 3600, VAXserver 3900

Cheetah.cfg (3100-96/3100-98)

```
#
# Copyright (C) 1999-2004 Software Resources International.
# All rights reserved.
#
# The software contained on this media is proprietary to and embodies
# the confidential technology of Software Resources International.
# Possession, use, duplication, or dissemination of the software and
# media is authorized only pursuant to a valid written license from
# Software Resources International.
#

#
# Sample configuration file for CHEETAH machines:
#
#  *  MicroVAX 3100 Model 96
#  *  MicroVAX 3100 Model 98
#

#
# Comment the following line if you do not want the log to be saved into
# file (change name of the file as well if you'd like). Each new session of
# the emulator appends its log to this file, therefore it grows bigger with
```

```
# time.
#

#set session log="charon.log"

#
# The following line tells the emulator where to preserve NVRAM content. It
# will keep the current time of the emulated VAX (when you do not run the
# emulator) and some (but not all!) of the console parameters.
#

#set toy container="charon.dat"

#
# The following line tells the emulator where to store intermediate state
# of the Flash ROM. It will keep the rest of console parameters. It is
# recommended to keep both previous and this line uncommented for the
# emulator to be able to correctly preserve the saved state of the console.
#

#set rom container="charon.rom"

#
# Disable or enable dynamic instruction translation by the cpu (DIT). The use
# of DIT may be also prohibited by the license. If not specified (i.e. when
# both lines remain commented out) the DIT is enabled as soon as the license
# allows to do so and is disabled otherwise ...
#

#set cpu dit_enable=false
#set cpu dit_enable=true

#
# Specify the size of RAM (default 16MB). Note that DIT (when enabled) also
# needs certain amount of memory which grows linearly following the size of
# memory specified here. Also remember that the dongle license might limit
# the maximum amount of memory.
#

#set ram size=16
#set ram size=128
```

```
#set ram size=512

#
# Now assign four built-in serial lines. Currently the emulator offers two
# possible ways of using built-in serial lines. First of them is connection
# to COM ports (via physical_serial_line). The second is to attach a third
# party terminal emulator (virtual_serial_line).
#

#load physical_serial_line TTA0 line="COMn:"
#load virtual_serial_line TTA0 port=10000
#load virtual_serial_line TTA0 port=10000 application="tta0.ht"
#set quart line[0]=TTA0

#load physical_serial_line TTA1 line="COM1:"
#load virtual_serial_line TTA1 port=10001
#load virtual_serial_line TTA1 port=10001 application="tta1.ht"
#set quart line[1]=TTA1

#load physical_serial_line TTA2 line="COM2:"
#load virtual_serial_line TTA2 port=10002
#load virtual_serial_line TTA2 port=10002 application="tta2.ht"
#set quart line[2]=TTA2

#load physical_serial_line OPA0 line="COMn:"
#load virtual_serial_line OPA0 port=10003
#load virtual_serial_line OPA0 port=10003 application="opa0.ht"
#set quart line[3]=OPA0

#
# Now assign few storage resources to built-in SCSI controllers. Currently it
# is capable of supporting raw SCSI devices (physical_scsi_device), virtual
# SCSI disks provided by DSK files (virtual_scsi_disk), physical floppy
# drives (floppy_scsi_device) and ATAPI CD-ROM's (atapi_scsi_device).
#

# connect a DSK file as disk to the PKA at SCSI ID 0
#load virtual_scsi_disk pka_0 scsi_bus=pka scsi_id=0
#set pka_0 container="<file-name>"

# connect physical SCSI device to the PKA at SCSI ID 1
```

```
#load physical_scsi_device pka_1 scsi_bus=pka scsi_id=1
#set pka_1 container="<device-name>"

# connect an ATAPI CDROM drive to the PKA at SCSI ID 6
#load atapi_scsi_device pka_6 scsi_bus=pka scsi_id=6
#set pka_6 container="\\.\CdRom0"

# connect the floppy disk drive to the PKB at SCSI ID 6
#load floppy_scsi_device pkb_6 scsi_bus=pkb scsi_id=6

#
# Uncomment one of the lines below to create a connection to the packet
driver
# which must be priory installed on the system. The first of the lines below
# connects through old-stile NDIS 3.0 packet driver. The second line requires
# new NDIS 5.0 packet driver. One of this connections is necessary for the
# built-in Ethernet Controller to be able to communicate to the LAN.
#

#load ndis_packet_port eza_0 interface="<device-name>"
#load ndis5_packet_port eza_0 interface="<device-name>"

#
# Once you have loaded one of EZA_0 packet interfaces, you may want to
connect
# the built-in Ethernet Controller EZA to this port.
#

#set eza interface=eza_0

#
# Load optional DHW42-AA (or DHW42-BA, or DHW42-CA) serial line controller
# (C-DAL).
#

#load DHW42AA/DHV11 TXA
#load DHW42BA/DHV11 TXA
#load DHW42CA/DHV11 TXA

#load physical_serial_line TXA0 line="COMn:"
#load virtual_serial_line TXA0 port=10010
```

```
#load virtual_serial_line TXA0 port=10010 application="txa0.ht"
#set TXA line[0]=TXA0

#load physical_serial_line TXA1 line="COMn:"
#load virtual_serial_line TXA1 port=10011
#load virtual_serial_line TXA1 port=10011 application="txa1.ht"
#set TXA line[1]=TXA1

#load physical_serial_line TXA2 line="COMn:"
#load virtual_serial_line TXA2 port=10012
#load virtual_serial_line TXA2 port=10012 application="txa2.ht"
#set TXA line[2]=TXA2

#load physical_serial_line TXA3 line="COMn:"
#load virtual_serial_line TXA3 port=10013
#load virtual_serial_line TXA3 port=10013 application="txa3.ht"
#set TXA line[3]=TXA3

#load physical_serial_line TXA4 line="COMn:"
#load virtual_serial_line TXA4 port=10014
#load virtual_serial_line TXA4 port=10014 application="txa4.ht"
#set TXA line[4]=TXA4

#load physical_serial_line TXA5 line="COMn:"
#load virtual_serial_line TXA5 port=10015
#load virtual_serial_line TXA5 port=10015 application="txa5.ht"
#set TXA line[5]=TXA5

#load physical_serial_line TXA6 line="COMn:"
#load virtual_serial_line TXA6 port=10016
#load virtual_serial_line TXA6 port=10016 application="txa6.ht"
#set TXA line[6]=TXA6

#load physical_serial_line TXA7 line="COMn:"
#load virtual_serial_line TXA7 port=10017
#load virtual_serial_line TXA7 port=10017 application="txa7.ht"
#set TXA line[7]=TXA7

# this is the end of the configuration file
##################################
```

Cheetah-q.cfg (4000-106/108)

```
#
# Copyright (C) 1999-2004 Software Resources International.
# All rights reserved.
#
# The software contained on this media is proprietary to and embodies
# the confidential technology of Software Resources International.
# Possession, use, duplication, or dissemination of the software and
# media is authorized only pursuant to a valid written license from
# Software Resources International.
#

#
# Sample configuration file for CHEETAH-Q machines:
#
#  *  VAX 4000 Model 106
#  *  VAX 4000 Model 108
#

#
# Comment the following line if you do not want the log to be saved into
# file (change name of the file as well if you'd like). Each new session of
# the emulator appends its log to this file, therefore it grows bigger with
# time.
#

#set session log="charon.log"

#
# The following line tells the emulator where to preserve NVRAM content. It
# will keep the current time of the emulated VAX (when you do not run the
# emulator) and some (but not all!) of the console parameters.
#

#set toy container="charon.dat"

#
# The following line tells the emulator where to store intermediate state
# of the Flash ROM. It will keep the rest of console parameters. It is
# recommended to keep both previous and this line uncommented for the
```

```
# emulator to be able to correctly preserve the saved state of the console.
#

#set rom container="charon.rom"

#
# Disable or enable dynamic instruction translation by the cpu (DIT). The use
# of DIT may be also prohibited by the license. If not specified (i.e. when
# both lines remain commented out) the DIT is enabled as soon as the license
# allows to do so and is disabled otherwise ...
#

#set cpu dit_enable=false
#set cpu dit_enable=true

#
# Specify the size of RAM (default 16MB). Note that DIT (when enabled) also
# needs certain amount of memory which grows linearly following the size of
# memory specified here. Also remember that the dongle license might limit
# the maximum amount of memory.
#

#set ram size=16
#set ram size=128
#set ram size=512

#
# Now assign four built-in serial lines. Currently the emulator offers two
# possible ways of using built-in serial lines. First of them is connection
# to COM ports (via physical_serial_line). The second is to attach a third
# party terminal emulator (virtual_serial_line).
#

#load physical_serial_line TTA0 line="COMn:"
#load virtual_serial_line TTA0 port=10000
#load virtual_serial_line TTA0 port=10000 application="tta0.ht"
#set quart line[0]=TTA0

#load physical_serial_line TTA1 line="COM1:"
#load virtual_serial_line TTA1 port=10001
#load virtual_serial_line TTA1 port=10001 application="tta1.ht"
```

```
#set quart line[1]=TTA1

#load physical_serial_line TTA2 line="COM2:"
#load virtual_serial_line TTA2 port=10002
#load virtual_serial_line TTA2 port=10002 application="tta2.ht"
#set quart line[2]=TTA2

#load physical_serial_line OPA0 line="COMn:"
#load virtual_serial_line OPA0 port=10003
#load virtual_serial_line OPA0 port=10003 application="opa0.ht"
#set quart line[3]=OPA0

#
# Now assign few storage resources to built-in SCSI controllers. Currently it
# is capable of supporting raw SCSI devices (physical_scsi_device), virtual
# SCSI disks provided by DSK files (virtual_scsi_disk), physical floppy
# drives (floppy_scsi_device) and ATAPI CD-ROM's (atapi_scsi_device).
#

# connect a DSK file as disk to the PKA at SCSI ID 0
#load virtual_scsi_disk pka_0 scsi_bus=pka scsi_id=0
#set pka_0 container="<file-name>"

# connect physical SCSI device to the PKA at SCSI ID 1
#load physical_scsi_device pka_1 scsi_bus=pka scsi_id=1
#set pka_1 container="<device-name>"

# connect an ATAPI CDROM drive to the PKA at SCSI ID 6
#load atapi_scsi_device pka_6 scsi_bus=pka scsi_id=6
#set pka_6 container="\\.\CdRom0"

# connect the floppy disk drive to the PKB at SCSI ID 6
#load floppy_scsi_device pkb_6 scsi_bus=pkb scsi_id=6

#
# Uncomment one of the lines below to create a connection to the packet
driver
# which must be priory installed on the system. The first of the lines below
# connects through old-stile NDIS 3.0 packet driver. The second line requires
# new NDIS 5.0 packet driver. One of this connections is necessary for the
# built-in Ethernet Controller to be able to communicate to the LAN.
```

```
#

#load ndis_packet_port eza_0 interface="<device-name>"
#load ndis5_packet_port eza_0 interface="<device-name>"

#
# Once you have loaded one of EZA_0 packet interfaces, you may want to
connect
# the built-in Ethernet Controller EZA to this port.
#

#set eza interface=eza_0

#
# Load optional DHW42-AA (or DHW42-BA, or DHW42-CA) serial line controller
# (C-DAL).
#

#load DHW42AA/DHV11 TXA
#load DHW42BA/DHV11 TXA
#load DHW42CA/DHV11 TXA

#load physical_serial_line TXA0 line="COMn:"
#load virtual_serial_line TXA0 port=10010
#load virtual_serial_line TXA0 port=10010 application="txa0.ht"
#set TXA line[0]=TXA0

#load physical_serial_line TXA1 line="COMn:"
#load virtual_serial_line TXA1 port=10011
#load virtual_serial_line TXA1 port=10011 application="txa1.ht"
#set TXA line[1]=TXA1

#load physical_serial_line TXA2 line="COMn:"
#load virtual_serial_line TXA2 port=10012
#load virtual_serial_line TXA2 port=10012 application="txa2.ht"
#set TXA line[2]=TXA2

#load physical_serial_line TXA3 line="COMn:"
#load virtual_serial_line TXA3 port=10013
#load virtual_serial_line TXA3 port=10013 application="txa3.ht"
#set TXA line[3]=TXA3
```

```
#load physical_serial_line TXA4 line="COMn:"
#load virtual_serial_line TXA4 port=10014
#load virtual_serial_line TXA4 port=10014 application="txa4.ht"
#set TXA line[4]=TXA4

#load physical_serial_line TXA5 line="COMn:"
#load virtual_serial_line TXA5 port=10015
#load virtual_serial_line TXA5 port=10015 application="txa5.ht"
#set TXA line[5]=TXA5

#load physical_serial_line TXA6 line="COMn:"
#load virtual_serial_line TXA6 port=10016
#load virtual_serial_line TXA6 port=10016 application="txa6.ht"
#set TXA line[6]=TXA6

#load physical_serial_line TXA7 line="COMn:"
#load virtual_serial_line TXA7 port=10017
#load virtual_serial_line TXA7 port=10017 application="txa7.ht"
#set TXA line[7]=TXA7

#
# Configure optional primary RQDX3 storage controller (MSCP/QBUS).
Handles disk
# images, disk drives, CD-ROM drives, magneto-optical drives, floppy drives.
#
# Containers to be specified as:
#
#  *  "<file-name>" for disk images;
#  *  "\\.\PhysicalDrive<N>" for local fixed disk drives;
#  *  "\\.\<device-letter>:" for removable disk drives (cdrom, optic, floppy)
#

#load RQDX3/RQDX3 DUA
#set DUA container[0]="..."
#set DUA container[1]="..."
#set DUA container[2]="..."
#set DUA container[255]="..."

#
# Configure optional secondary RQDX3 storage controller (MSCP/QBUS).
```

```
#

#load RQDX3/RQDX3 DUB address=017760334
#set DUB container[0]="..."
#set DUB container[1]="..."
#set DUB container[2]="..."
#set DUB container[255]="..."

#
# Configure optional primary TQK50 tape storage controller (TMSCP/QBUS).
Handles
# tape images and physical tape drives.
#
# Containers to be specified as:
#
#  *  "<file-name>" for tape images;
#  *  "\\.\Tape<N>" for local physical tape drives;
#

#load TQK50/TQK50 MUA
#set MUA container[0]="..."
#set MUA container[1]="..."
#set MUA container[2]="..."
#set MUA container[255]="..."

#
# Configure optional primary DELQA Ethernet adapter (QBUS).
#

#load DELQA/DEQNA XQA

#
# If you enabled XQA uncomment the lines below to create a connection to
the
# packet driver which must be priory installed on the system. The first line
# requires new NDIS 5.0 packet driver. This connection is necessary for the
# XQA to be able to communicate to the LAN.
#

#load ndis5_packet_port XQA0 interface="..."
#set XQA interface=XQA0
```

```
#
# Configure optional DZV11 serial line controller (QBUS). Address and vector
# must be set as required by operating system.
#

#load DZV11/DZ11 TTA
#set TTA address=... vector=...

#load physical_serial_line TTA0 line="COMn:"
#load virtual_serial_line TTA0 port=10110
#load virtual_serial_line TTA0 port=10110 application="tta0.ht"
#set TTA line[0]=TTA0

#load physical_serial_line TTA1 line="COMn:"
#load virtual_serial_line TTA1 port=10111
#load virtual_serial_line TTA1 port=10111 application="tta1.ht"
#set TTA line[1]=TTA1

#load physical_serial_line TTA2 line="COMn:"
#load virtual_serial_line TTA2 port=10112
#load virtual_serial_line TTA2 port=10112 application="tta2.ht"
#set TTA line[2]=TTA2

#load physical_serial_line TTA3 line="COMn:"
#load virtual_serial_line TTA3 port=10113
#load virtual_serial_line TTA3 port=10113 application="tta3.ht"
#set TTA line[3]=TTA3

#
# Configure optional DHV11 serial line controller (QBUS). Address and vector
# must be set as required by operating system.
#

#load DHV11/DHV11 TXA
#load DHQ11/DHV11 TXA
#load CXY08/DHV11 TXA
#load CXA16/DHV11 TXA
#load CXB16/DHV11 TXA
#set TXA address=... vector=...
```

```
#load physical_serial_line TXA0 line="COMn:"
#load virtual_serial_line TXA0 port=10010
#load virtual_serial_line TXA0 port=10010 application="txa0.ht"
#set TXA line[0]=TXA0

#load physical_serial_line TXA1 line="COMn:"
#load virtual_serial_line TXA1 port=10011
#load virtual_serial_line TXA1 port=10011 application="txa1.ht"
#set TXA line[1]=TXA1

#load physical_serial_line TXA2 line="COMn:"
#load virtual_serial_line TXA2 port=10012
#load virtual_serial_line TXA2 port=10012 application="txa2.ht"
#set TXA line[2]=TXA2

#load physical_serial_line TXA3 line="COMn:"
#load virtual_serial_line TXA3 port=10013
#load virtual_serial_line TXA3 port=10013 application="txa3.ht"
#set TXA line[3]=TXA3

#load physical_serial_line TXA4 line="COMn:"
#load virtual_serial_line TXA4 port=10014
#load virtual_serial_line TXA4 port=10014 application="txa4.ht"
#set TXA line[4]=TXA4

#load physical_serial_line TXA5 line="COMn:"
#load virtual_serial_line TXA5 port=10015
#load virtual_serial_line TXA5 port=10015 application="txa5.ht"
#set TXA line[5]=TXA5

#load physical_serial_line TXA6 line="COMn:"
#load virtual_serial_line TXA6 port=10016
#load virtual_serial_line TXA6 port=10016 application="txa6.ht"
#set TXA line[6]=TXA6

#load physical_serial_line TXA7 line="COMn:"
#load virtual_serial_line TXA7 port=10017
#load virtual_serial_line TXA7 port=10017 application="txa7.ht"
#set TXA line[7]=TXA7
```

```
# this is the end of the configuration file
################################
```

5.7 Mayfair.cfg (3600/3900)

```
#
# Copyright (C) 1999-2004 Software Resources International.
# All rights reserved.
#
# The software contained on this media is proprietary to and embodies
# the confidential technology of Software Resources International.
# Possession, use, duplication, or dissemination of the software and
# media is authorized only pursuant to a valid written license from
# Software Resources International.
#

#
# Sample configuration file for MAYFAIR machines:
#
#  *  MicroVAX 3600
#  *  MicroVAX 3900
#  *  VAXserver 3600
#  *  VAXserver 3900
#

#
# Comment the following line if you do not want the log to be saved into
# file (change name of the file as well if you'd like). Each new session of
# the emulator appends its log to this file, therefore it grows bigger with
# time.
#

#set session log="charon.log"

#
# The following line tells the emulator where to preserve NVRAM content. It
# will keep the current time of the emulated VAX (when you do not run the
# emulator) and console parameters (such as default boot device).
#
```

```
#set toy container="charon.dat"

#
# Disable or enable dynamic instruction translation by the cpu (DIT). The use
# of DIT may be also prohibited by the license. If not specified (i.e. when
# both lines remain commented out) the DIT is enabled as soon as the license
# allows to do so and is disabled otherwise ...
#

#set cpu dit_enable=false
#set cpu dit_enable=true

#
# Specify the size of RAM (default 16MB). Note that DIT (when enabled) also
# needs certain amount of memory which grows linearly following the size of
# memory specified here. Also remember that the dongle license might limit
# the maximum amount of memory.
#

#set ram size=16
#set ram size=32
#set ram size=64

#
# Now assign console built-in serial line. Currently the emulator offers two
# possible ways using serial lines. First of them is connection to COM ports
# (via physical_serial_line). The second is to attach a third party terminal
# emulator (virtual_serial_line).
#

#load physical_serial_line OPA0 line="COM1:"
#load virtual_serial_line OPA0 port=10003
#load virtual_serial_line OPA0 port=10003 application="opa0.ht"
#set uart line=OPA0

#
# Configure optional primary RQDX3 storage controller (MSCP/QBUS).
Handles disk
# images, disk drives, CD-ROM drives, magneto-optical drives, floppy drives.
#
# Containers to be specified as:
```

```
#
#  *  "<file-name>" for disk images;
#  *  "\\.\PhysicalDrive<N>" for local fixed disk drives;
#  *  "\\.\<device-letter>:" for removable disk drives (cdrom, optic, floppy)
#

#load RQDX3/RQDX3 DUA
#set DUA container[0]="..."
#set DUA container[1]="..."
#set DUA container[2]="..."
#set DUA container[255]="..."

#
# Configure optional secondary RQDX3 storage controller (MSCP/QBUS).
#

#load RQDX3/RQDX3 DUB address=017760334
#set DUB container[0]="..."
#set DUB container[1]="..."
#set DUB container[2]="..."
#set DUB container[255]="..."

#
# Configure optional primary TQK50 tape storage controller (TMSCP/QBUS).
Handles
# tape images and physical tape drives.
#
# Containers to be specified as:
#
#  *  "<file-name>" for tape images;
#  *  "\\.\Tape<N>" for local physical tape drives;
#

#load TQK50/TQK50 MUA
#set MUA container[0]="..."
#set MUA container[1]="..."
#set MUA container[2]="..."
#set MUA container[255]="..."

#
# Configure optional primary DELQA Ethernet adapter (QBUS).
```

```
#

#load DELQA/DEQNA XQA

#
# If you enabled XQA uncomment the lines below to create a connection to
the
# packet driver which must be priory installed on the system. The first line
# requires new NDIS 5.0 packet driver. This connection is necessary for the
# XQA to be able to communicate to the LAN.
#

#load ndis5_packet_port XQA0 interface="..."
#set XQA interface=XQA0

#
# Configure optional DZV11 serial line controller (QBUS). Address and vector
# must be set as required by operating system.
#

#load DZV11/DZ11 TTA
#set TTA address=... vector=...

#load physical_serial_line TTA0 line="COMn:"
#load virtual_serial_line TTA0 port=10110
#load virtual_serial_line TTA0 port=10110 application="tta0.ht"
#set TTA line[0]=TTA0

#load physical_serial_line TTA1 line="COMn:"
#load virtual_serial_line TTA1 port=10111
#load virtual_serial_line TTA1 port=10111 application="tta1.ht"
#set TTA line[1]=TTA1

#load physical_serial_line TTA2 line="COMn:"
#load virtual_serial_line TTA2 port=10112
#load virtual_serial_line TTA2 port=10112 application="tta2.ht"
#set TTA line[2]=TTA2

#load physical_serial_line TTA3 line="COMn:"
#load virtual_serial_line TTA3 port=10113
#load virtual_serial_line TTA3 port=10113 application="tta3.ht"
```

```
#set TTA line[3]=TTA3

#
# Configure optional DHV11 serial line controller (QBUS). Address and vector
# must be set as required by operating system.
#

#load DHV11/DHV11 TXA
#load DHQ11/DHV11 TXA
#load CXY08/DHV11 TXA
#load CXA16/DHV11 TXA
#load CXB16/DHV11 TXA
#set TXA address=... vector=...

#load physical_serial_line TXA0 line="COMn:"
#load virtual_serial_line TXA0 port=10010
#load virtual_serial_line TXA0 port=10010 application="txa0.ht"
#set TXA line[0]=TXA0

#load physical_serial_line TXA1 line="COMn:"
#load virtual_serial_line TXA1 port=10011
#load virtual_serial_line TXA1 port=10011 application="txa1.ht"
#set TXA line[1]=TXA1

#load physical_serial_line TXA2 line="COMn:"
#load virtual_serial_line TXA2 port=10012
#load virtual_serial_line TXA2 port=10012 application="txa2.ht"
#set TXA line[2]=TXA2

#load physical_serial_line TXA3 line="COMn:"
#load virtual_serial_line TXA3 port=10013
#load virtual_serial_line TXA3 port=10013 application="txa3.ht"
#set TXA line[3]=TXA3

#load physical_serial_line TXA4 line="COMn:"
#load virtual_serial_line TXA4 port=10014
#load virtual_serial_line TXA4 port=10014 application="txa4.ht"
#set TXA line[4]=TXA4

#load physical_serial_line TXA5 line="COMn:"
#load virtual_serial_line TXA5 port=10015
```

```
#load virtual_serial_line TXA5 port=10015 application="txa5.ht"
#set TXA line[5]=TXA5

#load physical_serial_line TXA6 line="COMn:"
#load virtual_serial_line TXA6 port=10016
#load virtual_serial_line TXA6 port=10016 application="txa6.ht"
#set TXA line[6]=TXA6

#load physical_serial_line TXA7 line="COMn:"
#load virtual_serial_line TXA7 port=10017
#load virtual_serial_line TXA7 port=10017 application="txa7.ht"
#set TXA line[7]=TXA7

# this is the end of the configuration file
##################################
```

Appendix B Pathworks 32

Compaq Pathworks32 is recommended as an ideal partner product for CHARON-VAX.

Compaq PATHWORKS 32 is a member of the PATHWORKS family of software products that provides PC connectivity in local and wide area networks. Windows 95, Windows 98, Windows NT 4.0, Windows 2000 and Windows XP PCs can be connected to the Compaq OpenVMS Operating System, the Compaq Tru64 UNIX Operating System, and other Windows 95, Windows 98, Windows NT, Windows 2000 and Windows XP systems and can use selected resources of those systems.

PATHWORKS 32 software layers on top of valid retail configurations of Windows 95, Windows 98, Windows NT 4.0, Windows 2000 and Windows XP and complements Microsoft networking software, so that different types of users can share information across the entire organization.

PATHWORKS 32 software ships on one software CD–ROM, which contains:

- Network Connectivity components
- eXcursion X Server
- PowerTerm 525 terminal emulator
- Online help and documentation PDF files in English, as well as in French, German, and Czech for selected PATHWORKS 32 components
- Localized user interfaces and documentation in French, German, and Czech for selected PATHWORKS 32 components.

CHARON-VAX customers may use Pathworks 32 as a File transfer system for PC to OpenVMS file transfers, an X-Windows emulator to replace their DEC workstation graphics or even an ideal terminal emulator to replace Digital VT Terminals. For details of the Compaq Pathworks32 see:

http://www.openvms.compaq.com/pathworks32/

Configuration steps

In VAX/VMS on CHARON-VAX/XM/XL for Windows:

Make sure that Charon's Packet Protocol (NDIS5) is installed and selected from the Windows Network Properties. If an other Internet Protocol (TCP/IP) has been enabled, disable it. (See 2.1.4, Network installation, for detailed instructions)

Start CHARON-VAX/XM/XL for Windows.

If VMS is used make sure that VAX/VMS contains :

- DECwindows Motif installed and licensed.
- TCP/IP Services installed, licensed and properly configured.
  - The XDM service is started using the **$ @SYS$MANAGER:TCPIP$CONFIG.COM**
  - To provide access to your host by eXcursion you need to copy and rename the file `XACCESS.TEMPLATE` to `XACCESS.TXT`. Edit the file `XACCESS.TXT` and check that the following line is uncommented:

    *** #any host can get a login window**.
- Restart VMS

Configure the eXcursion Control Panel

- In the XDMCP tab check the box Enable XDMCP

- In the XDMCP Startup Mode, choose Broadcast Mode and Max Host : 256
- Click on the Start Server Button

Now a List box appears and displays all XDMCP Hosts available.

For more information about eXcursion see the eXcursion User's Guide.

Appendix C FAQ and trouble shooting

Q: I have started CHARON-VAX/XM/XL as documented with no configuration file and nothing happens.

A: Start CHARON-VAX/XM/XL with the command: “CHARON CHARON.CFG” where CHARON.CFG is the name of your configuration file.

Q: When emulating a VAX with 512 MB of memory CHARON-VAX/XM/XL cannot allocate enough resources for the emulated memory.

A: Check that your host PC has sufficient memory to support your model of CHARON-VAX/XM/XL. See the section in this manual on host system requirements.

Q: I have previously used CHARON-VAX/Industrial can I use my existing configuration file with CHARON-VAX/XM/XL.

A: No, the configuration files for these two products are quite different. If in doubt on how to convert your configuration file contact your software supplier or Software Resources International.

Q: I am unsure about when and if geometry has to be specified for my disk images.

A: The geometry command was implemented for compatibility with early versions of VMS. CHARON-VAX/XM/XL version 1.2 and later will determine the disk geometry for real disks and disk images from the physical characteristics of the disk so you should not need you to specify the disk geometry.

Q: I have configured several disks on the same device of an emulated SCSI controller by using logical unit numbers, but the devices are not visible in VMS

A: Use the connect command in VMS as described in the chapter of this manual on CHARON-VAX/XM/XL disk drives.

Q: I cannot save the boot parameters of my emulated VAX from run to run.

A: Make sure you have defined files to hold the time of year (TOY) parameters and read only memory (ROM) parameters. Make sure that no corrupted versions of these files exist. If no file of that name exists CHARON-VAX will create the TOY or ROM container if required. Start the emulator and use the SRM console to set the boot parameters you need. Do not boot the operating system. Exit from the emulator and check that a ROM file has been saved. CHARON-VAX will only save a ROM file if you have changed the default parameters. Re-enter CHARON-VAX and check that the boot parameters you need have been saved. Boot the operating system and set the time. When you close down the operating system and exit the emulator a TOY container will be created. Please take care to give the emulator a few seconds save the details in the ROM and TOY containers. If in doubt delete the old ROM or TOY container file and start again. Once these files have been created correctly they will be maintained as required by CHARON-VAX and will not normally need to be re-created.

Q: My application uses the physical Ethernet address as a license key. It works on my original VAX, but not on the emulator.

A: Use the STATION_ADDRESS parameter to set the desired physical address on your emulated Ethernet card EZA.

Q: The emulator reports that an address is in use on the LAN. Example: "MPORT:734: Failed to submit RX I/O on 'EWB:' interface, (00-000000-9000) protocol. (8404) %SYSTEM-F-DEVINACT, device inactive".

A: There is another host on the LAN using the MAC address you specified in the controller's STATION_ADDRESS, or during DECnet configuration set by the VAX operating system. Resolution: specify a unique address for the controller's STATION_ADDRESS and DECnet address.

Q: The emulator warns you that your physical Ethernet connection is too fast. Example: "WARNING - 100 Mbps Network card or possibility to negotiate line speed over 10 Mbs has been detected on the 'EWB:' interface. The emulated VAX may not be able to maintain reliable Network operations over such a card. Review your configuration to ensure reliable Network performance. Refer to CHARON-VAX documentation and release notes for more detail."

A: Update: If you are running CHARON-VAX version 2 you should be able to achieve stable operations on a card set to 100 Mbs. You will still see the above warning but Network operations should be able to be maintained. If significant packet loss or an application malfunctions due to Network errors the first thing to try is setting the card down to 10 Mbs.

Reason: The network adapter assigned in the configuration file is configured with speed over 10 Mbps. It operates in auto-sense, 100Mbps or any other mode that can result in a connection speed over 10 Mbps. Hardware VAXes and CHARON-VAX/XM/XL version 1 did not support a Network speed in excess of 10 Mbps. If necessary configure your CHARON-VAX/XM/XL Network adapter to run at 10 Mbps.

Explanation: With a faster Network card, the CHARON-VAX/XM/XL packet port can deliver too many packets to the emulator, more than the VAX Operating System running under the emulator may be able to process. In that case the following can happen:

1. The receive queue grows, increasing the processing time for each packet as packets will wait in a queue. Even if a packet is not dropped it might introduce timeouts.

2. Some portion of the incoming traffic is lost "on the wire" (dropped due to queue overflow), which breaks some protocols. Note that even connection-oriented protocols with re-transition logic can be broken because the number of re-transition attempts can be exceeded.

Eventually VMS running under the emulator may disable the network card as result of repeated errors that will be seen as a network hang-up. Note that hardware VAX systems react the same way in case of traffic overload.

Appendix D VAX/VMS notes

Host system load and performance

VAX systems have an “idle” process that runs when the VAX operating system is waiting for the next activity. The means that all CPU time in the host system assigned to the CHARON-VAX/XM/XL application is used to execute MicroVAX instructions even when the VAX operating system is idle, resulting in a 100% CPU load. See the Chapter in Utilities titled CHARON CPU Usage Optimization for tips on how to reduce CHARON CPU usage.

CHARON-VAX/XM/XL is a Windows application, which will co-exist with other applications. The Windows scheduler handles the load distribution between applications, effectively allowing other (office) applications to operate normally if the host system is fast. Performance of complex configurations increases significantly on a Windows dual CPU system. The load distribution of the CHARON-VAX/XM/XL components across CPUs is automatic.

VAX/VMS clustering

An emulated VAX running on CHARON-VAX/XM/XL for Windows can be clustered together over the Ethernet, it may also join existing VAX or Alpha clusters. If planning a shared disk cluster please note that specific hardware is required and hardware iSCSI disks must be used, not disk images. Please contact us for more information. The following lines give the basic commands required to add your CHARON-VAX to and existing cluster

To add a CHARON-VAX/XM/XL to a cluster use the standard cluster licenses and commands. See the OpenVMS documentation for detailed instructions:

- OpenVMS Cluster Systems (AA-PV5WE-TK)
- Guidelines for OpenVMS Cluster Configurations (AA-Q28LE-TK)

Here is an example assuming that you are adding the emulated CHARON-VAX/XM/XL system to an existing system set up to receive an additional cluster member.

1. Add the cluster licenses to VAX OpenVMS

2. Rename the CHARON-VAX XM/XL disk to any unique name:
   $ SET VOLUME/LABEL=CHRVAX DUA0:
   $ PRODUCT REGISTER VOLUME OVMSVAXSYS DUA0:

3. Configure the VMS cluster on CHARON-VAX (in this example the system name is TST711):
   $ @CLUSTER_CONFIG
   Cluster Configuration Procedure
   Executing on a VAX System
   DECnet Phase IV is installed on this node.
   This procedure has detected that the LANACP
   LAN server process is running on this node.
   ....
   Do you want LANACP instead of DECnet for cluster boot serving? [YES] NO
   MAIN MENU
   1. ADD TST711 to existing cluster, or form a new cluster.
   2. MAKE a directory structure for a new root on a system disk.
   3. DELETE a root from a system disk.
   4. EXIT from this procedure.
    Enter choice [1]:
   ....
   Will the LAN be used for cluster communications (Y/N)? Y
   Enter this cluster's group number: 4000
   Will TST711 be a boot server [Y]? N
   Will TST711 be a disk server (Y/N)? Y
   Does this cluster contain a quorum disk [N]? N
   ....
   Do you want to run AUTOGEN now [Y]? N

4. Edit the file MODPARAMS.DAT:
   Change value of the parameter VOTES to 0.

5. Run the AUTOGEN utility
   $ @SYS$UPDATE:AUTOGEN GETDATA REBOOT

Index

Reader's Comment Form

We appreciate your comments, suggestions, criticism and updates of this manual. You can Email us your comments at:

vaxinfo@vaxemulator.com

Please refer to document number: 30-16-010

If you found any errors, please list them with their page number.